

***Títol: DISSENY I IMPLEMENTACIÓ D'UN SISTEMA DE
GESTIÓ DE PROMOCIONS FARMACÈUTIQUES***

Volum: 1

Alumne: CARLOS NARCISO MILLÁN

Director/Ponent: DANIELA TOST PARDELL

Departament: LSI

Data: Gener del 2009

DADES DEL PROJECTE

Títol del Projecte: DISSENY I IMPLEMENTACIÓ D'UN SISTEMA DE GESTIÓ DE PROMOCIONS FARMACÈUTIQUES

Nom de l'estudiant: CARLOS NARCISO MILLÁN

Titulació: ENGINYERIA TÈCNICA EN INFORMÀTICA DE GESTIÓ

Crèdits: 22,5

Director/Ponent: DANIELA TOST PARDELL

Departament: LSI

MEMBRES DEL TRIBUNAL (nom i signatura)

President: LUIS SOLANO ALBAJES

Vocal: JOSÉ ANTONIO GONZÁLEZ ALAUSTRE

Secretari: DANIELA TOST PARDELL

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

*A mi familia que siempre creyó en mi,
a mis amigos por aguantarme,
esto es por ti abuelo, gracias.*

1 • Introducción 10

1.1	Preámbulo	11
1.2	Descripción	11
1.3	Estado inicial	14
1.4	Objetivos del proyecto	14
1.5	Planificación inicial	17

2 • Modelo de requerimientos 20

2.1	Análisis previo	21
2.2	Etapas	21
2.2.1	Obtener	21
2.2.2	Validar	23
2.2.3	Contrastar	24
2.3	Requerimientos funcionales	25
2.3.1	Buscador de colaboraciones	25
2.3.2	User queries	25
2.3.3	System Queries	25
2.3.4	Quick Launch	26
2.3.5	Vista rápida	26
2.3.6	Mantenimiento de colaboraciones	26
2.3.7	Cambios de estado	26
2.3.8	Auto salvado	26
2.3.9	Última conexión	27
2.3.10	Definición de eventos	27
2.3.11	Tipologías	27
2.4	Requerimientos no funcionales	27
2.4.1	Accesible	27
2.4.2	Seguridad	27

2.4.3 Requisito de Interfaz de usuario	28
2.4.4 Mantenible	29
2.4.5 Eficiente	29
2.4.6 Fiable	30
2.4.7 Escalable	30
2.4.8 Tecnología	30
2.4.9 Requisito de documentación	30

3 • Análisis 33

3.1 Casos de uso	34
3.1.1 Visión general	34
3.1.2 Roles	35
3.1.3 Visibilidad	35
3.1.4 Especificaciones	36
3.1.4.1 Gestión de colaboraciones	36
3.1.4.2 Gestión de eventos	46
3.1.4.3 Gestión de tipologías	51
3.2 Modelo conceptual	60
3.2.1 Colaboración	60
3.2.2 Evento y definición de evento	60
3.2.3 Costes	61
3.2.4 Conceptos	61
3.2.5 Coste - Concepto	62
3.2.6 Tipología	63
3.2.7 Producto	64
3.2.8 Perceptor	64
3.2.9 Perceptor - Coste	65
3.2.10 Budget Owner	66
3.2.11 Budget Owner - Coste	67
3.2.12 Estados	68

4	• Implementación y pruebas	71
4.1	Diseño	72
4.1.1	Arquitectura cliente-servidor	73
4.1.1.1	Características de un cliente	75
4.1.1.2	Características de un servidor	75
4.1.1.3	Comparativa con otras arquitecturas	76
4.1.2	Arquitectura de Software	79
4.1.2.1	Capas y niveles	80
4.1.2.2	Justificación	81
4.1.3	Metodología de diseño	82
4.1.3.1	Justificación	83
4.1.4	Estudio tecnológico	83
4.1.4.1	Tecnologías y herramientas utilizadas	84
4.1.4.1.1	Plataforma de programación	84
4.1.4.1.2	Servidor Web	95
4.2	Implementación	100
4.2.1	Vistas del sistema	100
4.2.1.1	Menús de navegación	104
4.2.1.2	Buscador de colaboraciones	105
4.2.1.3	Quick launch	106
4.2.1.4	Crear colaboración	107
4.2.1.5	Gestión de colaboraciones	108
4.2.1.6	Gestión de definiciones de evento	109
4.2.1.7	Gestión de Tipologías	110
4.3	Pruebas	113
4.3.1	Pruebas unitarias	115
4.3.2	Pruebas de integración	115
4.3.3	Pruebas de validación	115
4.3.4	Test de usabilidad	116
4.3.4.1	Proceso	117
4.3.4.2	Resultados y conclusiones	122

5 • Estudio temporal y económico 127

5.1 Estudio temporal inicial	128
5.2 Estudio temporal real	132
5.3 Desviación temporal	134
5.4 Estudio económico	135
5.4.1 Recursos materiales	135
5.4.2 Recursos humanos	136
5.4.3 Gastos totales	137

6 • Conclusiones 139

6.1 La aplicación	140
6.2 A nivel personal	141

7 • Bibliografía 143

7.1 Internet	144
7.2 Libros de texto	145

8	• Anexo-I: Manual de usuario	147
8.1	Introducción	148
8.2	Colaboraciones	149
8.3	Estado de las colaboraciones	150
8.4	Proceso de las colaboraciones	151
8.5	Grid de colaboraciones	152
8.6	Uso de filtros	154
8.7	Crear una colaboración	158
8.7.1	Tipo Otros	158
8.7.2	Tipo Evento	162
8.8	Operaciones	165
8.8.1	Operaciones sobre colaboraciones inscritas	165
8.8.2	Operaciones sobre el Grid	166
8.9	Definición de eventos	168

1 ● Introducción

En este bloque presentaremos la motivación y objetivos que pretende alcanzar este proyecto. Resulta imprescindible para poder adquirir una idea general del proyecto y así poder entender mejor los siguientes capítulos.

Contenidos de la sección

- 1.1 Preámbulo*
- 1.2 Descripción*
- 1.3 Estado inicial*
- 1.4 Objetivos*
- 1.5 Planificación*

1.1 Preámbulo

Antes de comenzar la lectura es importante presentar al lector un par de temas importantes.

El proyecto se realiza en condición de empleado y en colaboración con la empresa Grubit, la cual ofertó este proyecto a la empresa farmacéutica en cuestión. Este proyecto inicial se dividió en tres módulos (Acuerdos, Colaboraciones y Administración), el PFC tratará el módulo de colaboraciones del cual fui responsable.

Mis responsabilidades en este proyecto englobaron la totalidad de la construcción del sistema: toma de requerimientos, especificación, diseño y implementación del mismo.

1.2 Descripción

El proyecto surge con motivo de satisfacer la necesidad de una empresa farmacéutica en gestionar colaboraciones y acuerdos realizados con los profesionales de los hospitales. Ante la inexistencia de una herramienta capaz de gestionar esta información deciden construir una herramienta a medida.

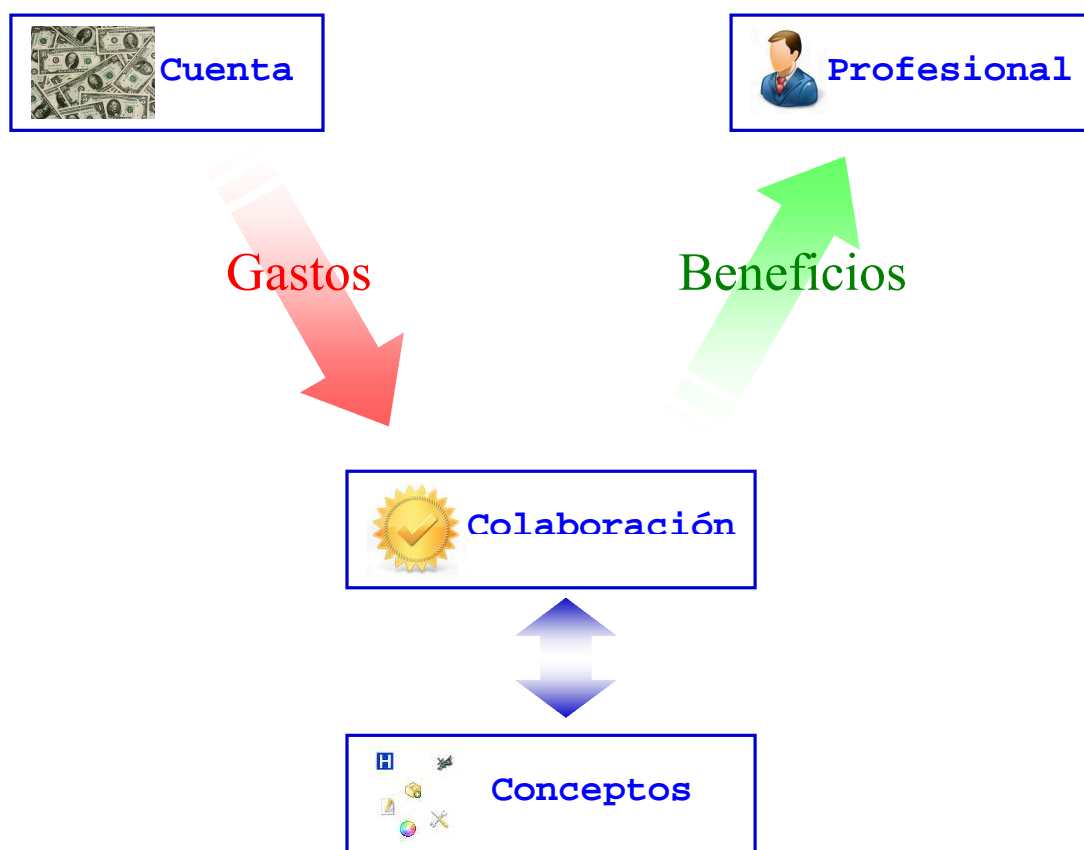
La empresa tiene diversos profesionales encargados de representar sus productos en diversos hospitales de España. Estos son los llamados representantes. Ellos son los encargados de realizar una serie de colaboraciones con los profesionales de los hospitales a su cargo.

Cada colaboración tiene asignado algunos de los productos que la empresa quiere promocionar.

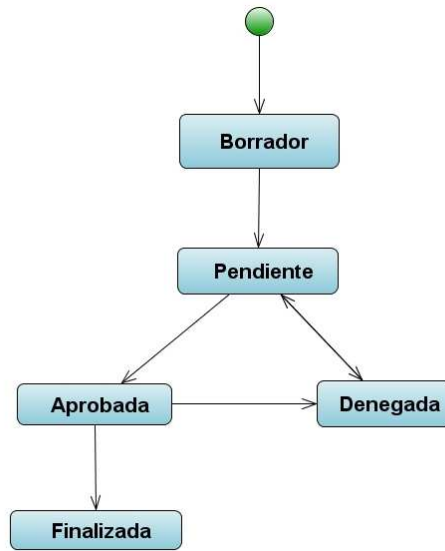
Una colaboración es un elemento consistente en un conjunto de beneficios a los que llamaremos conceptos. Estos conceptos pueden ser un viaje, un hotel, una donación...

En ocasiones las colaboraciones se agrupan en eventos. Todas las colaboraciones agrupadas en un mismo evento comparten una serie de características.

Los conceptos de las colaboraciones representan un coste para la empresa. Dicho coste será asumido por una determinada cuenta, de la cual existirá una persona responsable.



Las colaboraciones puede pasar por diversos estados, una vez creada podrá ser aprobada o rechazada por la persona encargada de supervisarlas. Tras su aprobación la colaboración termina por finalizar en el momento en que el supervisor lo indica.



Esta es, a grandes rasgos, la descripción del problema que queremos gestionar mediante una aplicación informática.

1.3 Estado inicial

La empresa no dispone de ningún sistema informático que integre todo el proceso de creación y evolución de las colaboraciones.

No existe una base de datos global que integre todos los datos necesarios, estos se encuentran almacenados de forma estática y su acceso es complejo. El proceso de aprobación de las colaboraciones resulta lento y costoso.

No existe ningún producto que sea capaz de cubrir las necesidades de gestión de las colaboraciones. Por eso acuden a Grubit, para que les plantee una solución.

Partimos, por tanto, de una situación inicial perfecta para iniciar un sistema desde cero que unifique y gestione los datos que se encuentran dispersos en distintos formatos y ubicaciones.

1.4 Objetivos del proyecto

El principal objetivo del proyecto es la creación de un sistema de información capaz de gestionar los datos correspondientes a las colaboraciones con los profesionales de los distintos hospitales con los que tiene trato.

Los objetivos que deberá asumir este proyecto son:

Diseño de una arquitectura del software capaz de almacenar y gestionar los datos de las colaboraciones de manera segura y eficiente.

Este es el objetivo inicial y más importante, ya que de las decisiones que se extraigan en este punto dependerá en gran medida la solidez del sistema que queremos construir.

Deberemos decidir las tecnologías adecuadas para la construcción del sistema, diseñar el software y el sistema de almacenaje de la información.

Diseño y implementación de una herramienta de búsqueda.

Los empleados de la farmacéutica encargados de la gestión de las colaboraciones requieren de una herramienta capaz de acceder de una manera rápida a cualquiera de las mismas. Esta herramienta ha de ser capaz de buscar por cualquier dato referente a la colaboración. Requieren la posibilidad de crear filtros de búsqueda personalizados para el acceso a estos datos.

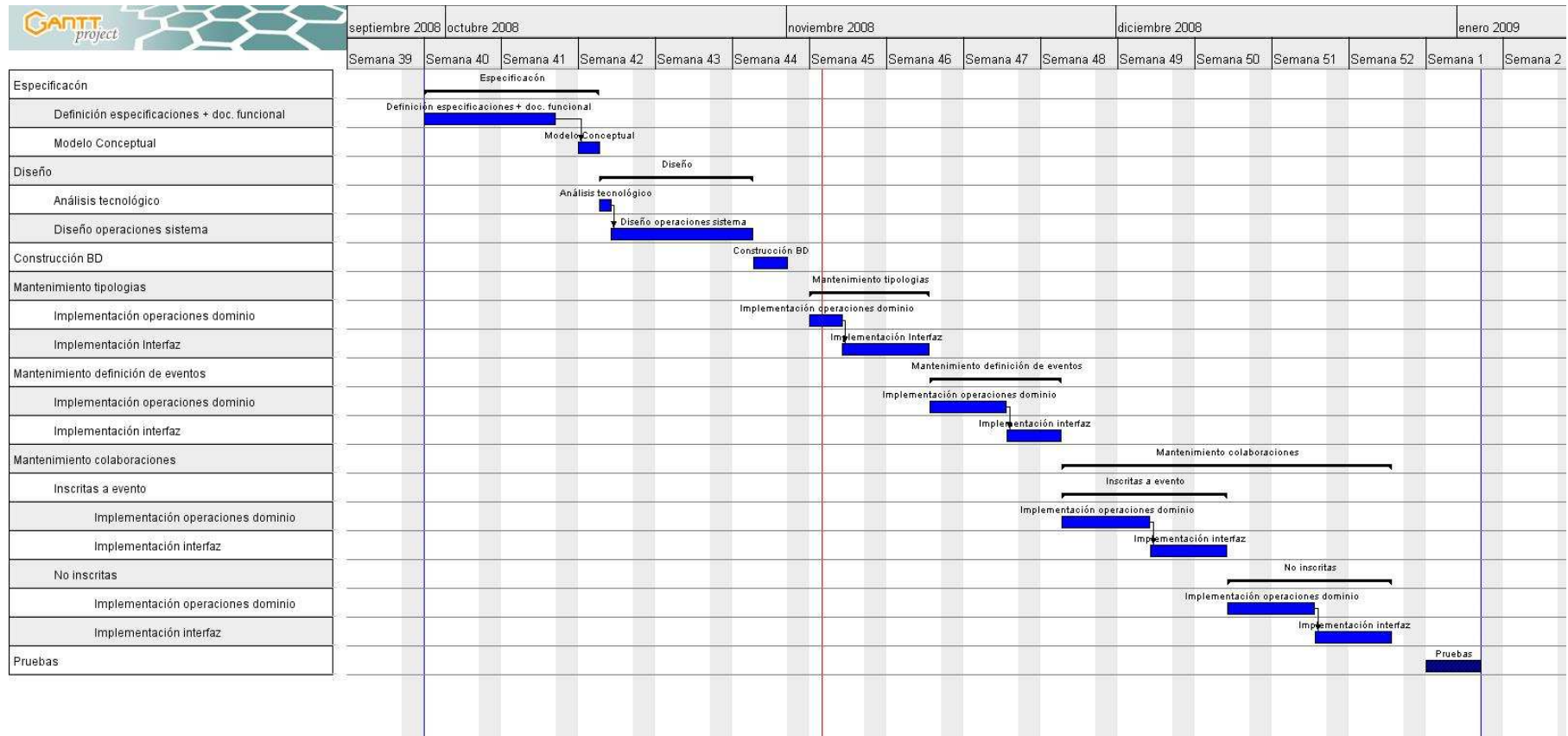
Diseño y implementación de las herramientas para la gestión de las colaboraciones.

Los empleados de la empresa deberán poder visualizar los datos de las colaboraciones. Deben poder también crearlas, modificarlas o borrarlas. Estas acciones estarán limitadas según el rol de usuario con el que accedan al sistema. El sistema deberá disponer de las herramientas necesarias para cubrir estos puntos.

Diseño y implementación de herramientas de administración de las colaboraciones

Un usuario podrá administrar características de las colaboraciones tales como la *tipología* o la *definición de eventos*. Para ello la aplicación deberá constar de herramientas que las gestionen.

1.5 Planificación inicial



Nombre	Fecha de inicio	Fecha de fin	Duración
[-] Especificación	29/09/08	15/10/08	12
Definición especificaciones + doc. funcional	29/09/08	11/10/08	10
Modelo Conceptual	13/10/08	15/10/08	2
[-] Diseño	15/10/08	29/10/08	10
Análisis tecnológico	15/10/08	16/10/08	1
Diseño operaciones sistema	16/10/08	29/10/08	9
Construcción BD	29/10/08	1/11/08	3
[-] Mantenimiento tipologías	3/11/08	14/11/08	9
Implementación operaciones dominio	3/11/08	6/11/08	3
Implementación Interfaz	6/11/08	14/11/08	6
[-] Mantenimiento definición de eventos	14/11/08	26/11/08	8
Implementación operaciones dominio	14/11/08	21/11/08	5
Implementación interfaz	21/11/08	26/11/08	3
[-] Mantenimiento colaboraciones	26/11/08	26/12/08	22
[-] Inscritas a evento	26/11/08	11/12/08	11
Implementación operaciones dominio	26/11/08	4/12/08	6
Implementación interfaz	4/12/08	11/12/08	5
[-] No inscritas	11/12/08	26/12/08	11
Implementación operaciones dominio	11/12/08	19/12/08	6
Implementación interfaz	19/12/08	26/12/08	5
Pruebas	29/12/08	3/01/09	5

2 • Modelo de requerimientos

En este capítulo realizaremos un análisis de la situación actual y extraeremos los requerimientos funcionales y no funcionales que conformarán la aplicación a construir.

Contenidos de la sección

- 2.1 Análisis previo*
- 2.2 Metodología seguida*
- 2.3 Requerimientos funcionales*
- 2.4 Requerimientos no funcionales*

2.1 Análisis previo

La empresa parte de una situación inicial en la que la gestión de las colaboraciones se realiza de manera muy rudimentaria.

Los responsables de la empresa tienen muy claro que funcionalidades deberá ser capaz de realizar la herramienta.

El objetivo principal de los requisitos es concretar, todo aquello que los clientes exigen al sistema y todo lo que los desarrolladores necesitan para construirlo.

2.2 Etapas

La extracción de estos requisitos para un sistema requiere de un seguido de etapas que se detallan a continuación.

2.2.1 Obtener

La primera de las etapas a las que nos enfrentaremos es a la de obtención de requisitos. En ella, se pretende, generar todos los requisitos posibles exigidos por el cliente. En este proceso, es de vital importancia tener contacto con el cliente, ya que nunca se debe olvidar que el sistema software estará destinado para él.

Según el tipo de sistema software, es muy probable, que también sea necesario tener contacto con el usuario final del sistema software que se debe desarrollar, ya que el usuario final, no tiene por que ser el cliente. Incluso, en

otros sistemas software también es posible, que se deba tener contacto con distribuidores.

En definitiva, que en la fase de obtención de requisitos, de algunos sistemas, es muy probable, que sea necesario mantener contacto con todas aquellas personas que puedan verse directa o indirectamente, implicadas por el sistema software que se va a desarrollar.

Para poder generar un conjunto de requisitos, existen varias metodologías entre las que se encuentran el prototipaje, las encuestas, las entrevistas y las reuniones en grupo.

La metodología seguida para el análisis de requerimientos es clave, ya que de ella depende todo el desarrollo del sistema y que ésta cumpla los objetivos previstos. Por ese motivo se hizo especial hincapié en la estrategia de toma de requerimientos ha seguir.

Tras estudiar distintas posibilidades se optó por utilizar una estrategia que incluyera varias metodologías, de esta manera aprovecharíamos las ventajas de cada una de ellas y minimizaríamos sus limitaciones.

A continuación se detallan las metodologías seguidas.

Entrevistas

Esta fue la primera de las metodologías usadas y por lo tanto en la que más volumen de requerimientos se esperaba extraer.

Las entrevistas son un método común. Esta estrategia no consiste en entrevistar a toda la gente que se relacionará con el sistema, sino a una selección de personas que represente a todos los sectores críticos de la

organización, con el énfasis puesto en los sectores más afectados o que harán un uso más frecuente del nuevo sistema.

Los requerimientos que surgieron en estas entrevistas a menudo se contradecían unos a otros o se formulaban desde la ignorancia de los detalles del funcionamiento del sistema.

Prototipos

Con tal de terminar de perfilar los requerimientos del sistema, se prepararon pequeñas muestras, de funcionalidad limitada, de cómo sería el producto una vez terminada. Estos prototipos ayudaron a conocer la opinión de los usuarios, haciendo rectificar algunos de los requisitos tomados.

Los prototipos ayudaron principalmente a las decisiones del diseño y de la interfaz de usuario. Sin embargo, no proporcionaron nuevos requisitos.

2.2.2 Validar

Una vez extraída la lista de requerimientos, la siguiente etapa consiste en validarlos. Esta etapa tendrá como input, el output del proceso de obtención de requisitos, es decir el listado de requisitos del sistema.

En éste proceso algunos de los requisitos de la lista de requisitos del sistema, se validan y otros se anulan, y de los que se validan, normalmente se hace un ranking, discriminando según su importancia, ya que es normal que algunos requisitos sean más importantes para el cliente que otros.

Del proceso de validación de requisitos se genera un listado de requisitos validados del sistema, que es un subconjunto de los requisitos obtenidos en el proceso de obtención de requisitos.

2.2.3 Contrastar

El tercero y último de los procesos es el de contrastar los requisitos. Éste requiere, que antes se haya efectuado el proceso de validación de requisitos, ya que éste proceso tiene como input, el output del proceso de validación, es decir el listado de requisitos validados del sistema.

En el proceso de contrastar los requisitos, se produce una interacción con las personas implicadas con el sistema software a desarrollar.

En éste proceso se analizan y contrastan cada uno de los requisitos del listado generado en el proceso de validación.

En función del ranking de los requisitos validados, y de las opiniones obtenidas de las personas implicadas en el sistema software, algunos de los requisitos validados se confirman como correctos, y otros se eliminan del listado ya que generan inconsistencias.

Del proceso de contrastar los requisitos, se genera un listado de requisitos correctos del sistema, que es un subconjunto de los requisitos obtenidos en el proceso de validación.

Del output producido por el proceso de contrastar los requisitos, se obtiene el listado de requisitos correctos del sistema.

A continuación dividiremos los requerimientos extraídos en dos grandes grupos, según sean funcionales o no.

2.3 Requerimientos funcionales

Los requerimientos funcionales son aquellas funcionalidades que el sistema ha de ofrecer a sus usuarios.

Estos son:

2.3.1 Buscador de colaboraciones

La aplicación deberá contener una herramienta consistente en un grid paginado en el que se mostrarán las colaboraciones. Se incorporará una herramienta de búsqueda que permita discriminar por cualquier campo mostrado en el grid. Se mostrarán un máximo de diez registros en cada una de las páginas.

2.3.2 User queries

La aplicación debe permitir que los usuarios puedan crear consultas y almacenarlas. La aplicación mostrará al usuario conectado sus consultas almacenadas, al seleccionarla será ejecutada.

2.3.3 System Queries

La aplicación ofrecerá una lista de consultas predeterminadas que el usuario podrá seleccionar para su ejecución.

2.3.4 Quick Launch




Se ofrecerá una herramienta de acceso rápido al mantenimiento y ejecución de *User Queries* y *System Queries*

2.3.5 Vista rápida

Al seleccionar algún registro de grid, la aplicación mostrará sus detalles.

2.3.6 Mantenimiento de colaboraciones

La aplicación deberá contener las siguientes herramientas de mantenimiento de las colaboraciones.

-  Crear
-  Borrar
-  Copiar

2.3.7 Cambios de estado

Las colaboraciones pasarán por una serie de estados, estos son: borrador, pendiente, aprobada, denegada y finalizada.

Se permitirán las siguientes acciones sobre las colaboraciones:

-  Solicitar
-  Aprobar
-  Denegar
-  Finalizar

2.3.8 Auto salvado

El sistema almacenará de manera automática los cambios producidos en una colaboración durante su edición. El auto salvado se pondrá en marcha en el momento en que el usuario seccione el preceptor de la colaboración.

2.3.9 Última conexión

El sistema almacenará la fecha de última conexión del usuario.

2.3.10 Definición de eventos

La aplicación tendrá una herramienta capaz de gestionar las definiciones de eventos.

2.3.11 Tipologías

Se requiere la existencia de una herramienta que gestione las tipologías de las colaboraciones.

2.4 Requerimientos no funcionales

Los requisitos no funcionales, son aquellos que describen las exigencias que pueden provocar una limitación en el sistema.

2.4.1 Accesible

Los usuarios de la aplicación podrán acceder mediante el acceso a un servidor web a través de internet.

2.4.2 Seguridad

Proteger los datos que están almacenados en el sistema y restringir el acceso a la información.

La seguridad es un aspecto muy importante, que se debe tener en cuenta en cualquier sistema informático. Pero en el caso de las aplicaciones Web el problema se acentúa ya que se podrá acceder desde cualquier ordenador que este conectado a Internet.

Además, hay que tener en cuenta que toda la información con la que los usuarios completan los formularios de la aplicación, al ser enviada, circula por la red.

Por estos motivos, el sistema deberá garantizar la máxima seguridad, tanto desde el punto de vista del acceso a la aplicación como de la integridad de la información almacenada.

Para alcanzar este grado de seguridad nuestro sistema utilizará los siguientes métodos.

Acceso mediante usuario y contraseña: Cualquier usuario que desee utilizar la aplicación deberá previamente identificarse mediante un nombre de usuario y una contraseña. Se requiere que la validación de usuario y contraseña se realice a través del sistema operativo (...) No se permitirá el acceso de usuarios anónimos a la aplicación.

Configuración segura del servidor Web: Al tratarse de una aplicación Web que se ejecuta sobre un servidor Web, deberá garantizarse que la configuración de seguridad de dicho servidor es lo suficientemente robusta como para evitar intrusiones y ataques no deseados al sistema.

2.4.3 Requisito de Interfaz de usuario

La interfaz de usuario deberá presentar un aspecto agradable y deberá ser intuitiva para que usuarios con poca experiencia puedan adaptarse rápidamente a su utilización.

La interfaz de usuario será lo más similar posible a las aplicaciones clásicas de escritorio. Esto implica minimizar

al máximo las recargas del navegador, así como incluir en las interfaces componentes tales como menús o barras de herramientas

2.4.4 Mantenible

Este requisito, hace referencia la dificultad que supone realizar posibles modificaciones y extensiones del sistema.

Existen varios motivos por el cual nuestro sistema podría sufrirlas.

Ya sea para corregir, mejorar o adaptar el software. O porque surja o cambie alguna necesidad. El sistema deberá tener muy en cuenta este requisito ya que es bastante probable que suceda alguno de los motivos citado

Así que para desarrollar el sistema seguiremos una metodología para construir un sistema flexible a los cambios y tendremos en cuenta los criterios que condicionan ésta flexibilidad.

2.4.5 Eficiente

La eficiencia es la capacidad del producto de software para proporcionar un funcionamiento apropiado, utilizando una serie de recursos y bajo unas condiciones indicadas.

Este criterio de calidad será evaluado en función de los siguientes criterios.

El tiempo de funcionamiento: Es decir, según el tiempo que tarde en proporcionar respuestas correctas, el tiempo que tarde en procesar información, etc.

La utilización de recursos: Este criterio valorará el número y el tipo de recursos que necesitará utilizar el sistema para que funcione correctamente.

El sistema deberá ser lo más eficiente posible, consiguiendo una sensación de rapidez de cara al usuario que facilite el trabajo con la herramienta.

El tiempo de respuesta de las operaciones, el rendimiento y la utilización de los recursos ha de ser óptima.

2.4.6 Fiable

Tiene que tratarse de un sistema tolerante a los fallos, robusto y con un buen tratamiento de errores.

2.4.7 Escalable

Tiene que poder soportar un incremento considerable de peticiones sin ver afectado su rendimiento.

2.4.8 Tecnología

El sistema debe funcionar sobre un servidor de aplicaciones Tomcat. Este servidor estará ubicado en las dependencias de la compañía y no será de uso exclusivo de nuestro sistema. Esto implica que otras aplicaciones estarán en funcionamiento en el mismo servidor que el nuestro.

2.4.9 Requisito de documentación

Se deberá elaborar un breve manual de usuario, con la información operacional que describa como reproducir todas las funcionales del sistema.

Este manual de usuario, tendrá que facilitar los conocimientos necesarios para poder usar correctamente la aplicación.

Debe tener un nivel de detalle elevado, y tendrá imágenes que ilustren las explicaciones y faciliten al usuario su aprendizaje.

3 • Análisis

A partir de los requerimientos obtenidos en el capítulo anterior especificaremos los casos de uso. También realizaremos una descripción detallada del modelo conceptual que seguirá nuestro sistema.

Contenidos de la sección

- 3.1 Casos de uso*
- 3.2 Descripción detallada*
- 3.3 Modelo conceptual*

3.1 Casos de uso

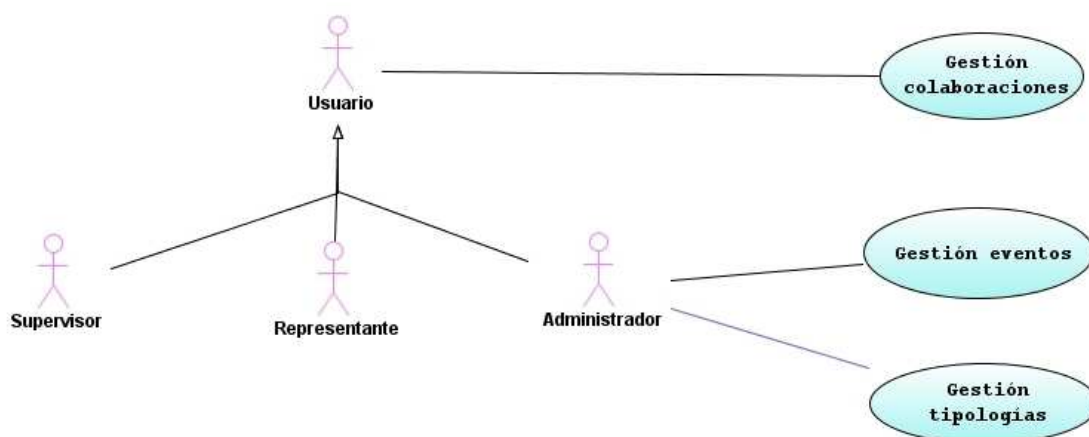
En este apartado realizaremos una descripción de alto nivel de las funcionalidades del sistema.

Para realizar esta descripción usaremos los casos de uso, los cuales describen a los actores utilizando un sistema con el fin de satisfacer un objetivo

Durante esta descripción se evitarán conceptos de diseño e implementación, ya que las funcionalidades del sistema no deben depender de los mismos.

3.1.1 Visión general

Después de ver los requerimientos funcionales del sistema, hemos detallado un seguido de casos de uso específicos que agruparemos en tres casos de uso más generales. Estos casos de uso son: Gestión de Colaboraciones, Gestión de Eventos y Gestión de Tipologías.



3.1.2 Roles

También, se han definido diferentes roles para los usuarios del sistema, ya que dependiendo del tipo de usuario, estos tendrán unos privilegios u otros. Los roles definidos para nuestro sistema son:

- **Representante:** Un representante es el agente encargado de llegar a acuerdos con los empleados de los centros que tenga asignados.
- **Supervisor:** Un supervisor es el agente encargado de gestionar y supervisar los acuerdos a los que llega el representante.
- **Administrador:** Gestionará y tendrá acceso a todo el sistema.

3.1.3 Visibilidad

La visibilidad de las colaboraciones viene definida por los centros que tiene asignados su usuario. Es decir, cada usuario solo verá las colaboraciones de los centros que tiene asignado.

Un administrador tiene asignados todos los centros existentes en el sistema.

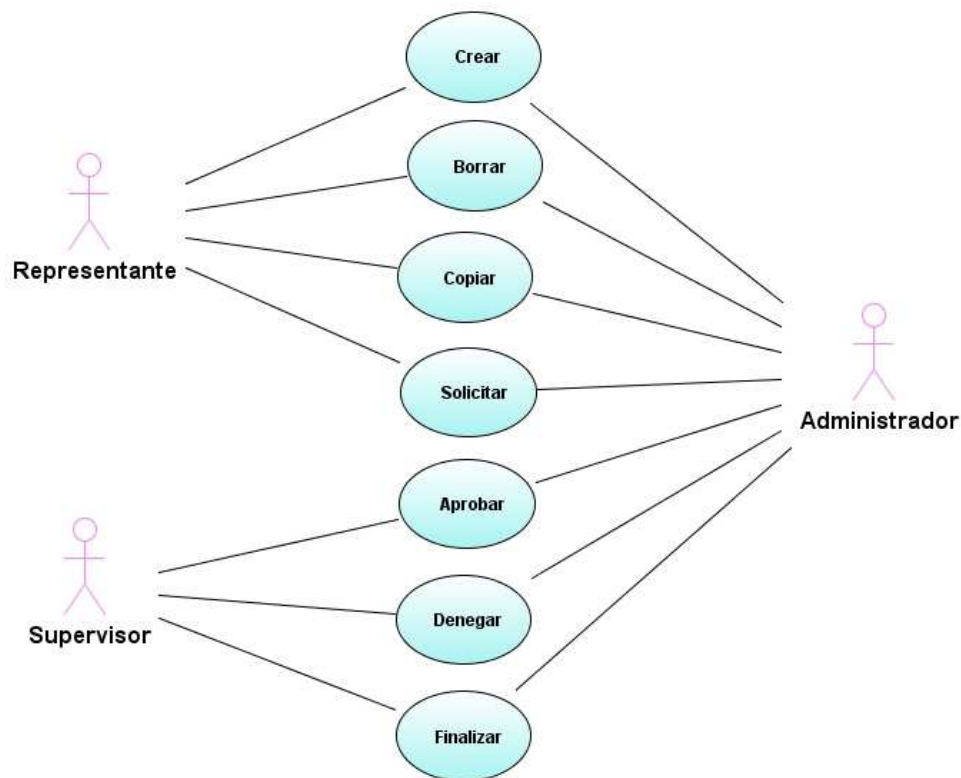
3.1.4 Especificaciones

3.1.4.1 Gestión de colaboraciones

Caso de uso: Gestión de colaboraciones

Actores: Representante, Supervisor, Administrador.

Descripción: Esta funcionalidad permite gestionar todas las posibles acciones que se pueden realizar sobre una colaboración, ya sea creando, eliminando o modificándolo; por este motivo se podría decir que encapsula un seguido de casos de uso más específicos que especificaremos a continuación. Estos casos de uso son *crear, borrar, copiar, solicitar, aprobar, denegar y finalizar*.



Caso de uso: Crear colaboración.

Actores: Representante, Administrador.

Propósito: Dar de alta una colaboración en el sistema.

Resumen: La colaboración será dada de alta en el sistema con los datos que haya introducido.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona la opción 'new sponsorship'	2.- El sistema muestra una lista con los tipos de colaboración que se pueden crear dividida en dos, una para colaboraciones tipo <i>otros</i> y otra para tipo <i>eventos</i> .
3.- El usuario selecciona el tipo de colaboración que quiere crear.	4.- El sistema muestra el interfaz adecuado para el tipo de colaboración seccionada.
5.- El usuario inserta los datos en el formulario.	6.- En el momento que el usuario inserta el <i>perceptor</i> en el formulario se activa el <i>auto salvado</i> , el cual almacena los datos del formulario ya insertados. A partir de ese momento todos los datos que se vayan insertando serán automáticamente almacenados.
7.- El usuario hace clic al botón <i>Exit</i>	8.- El sistema muestra un diálogo

<p>9.- El usuario selecciona que <u>sí</u> quiere salvar los datos.</p>	<p>de opción preguntando si quiere salvar los datos</p> <p>10.- El sistema valida los datos introducidos.</p> <p>11.- Si los datos son correctos el sistema sale del formulario de edición.</p>
---	---

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
<p>9.- El usuario selecciona que <u>no</u> quiere guardar los datos</p>	<p>10.- El sistema sale del formulario de edición.</p>

Curso alternativo 2

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
<p>12.- El usuario edita los datos del formulario y hace clic al botón <i>Exit</i></p>	<p>11.- Si los datos no son correctos el sistema muestra un diálogo con los errores detectados.</p> <p>13.- Volvemos al punto 8 del curso típico.</p>

Caso de uso: Borrar colaboración.

Actores: Representante, Administrador.

Propósito: Borrar una colaboración.

Resumen: La colaboración será borrada del sistema tras confirmación del usuario.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración y pulsa la opción 'delete sponsorship'	2.- El sistema muestra un mensaje confirmando la baja de la colaboración
3.- El usuario confirma su decisión de borrar la colaboración.	4.- El sistema elimina toda la información asociada con la colaboración que quiere eliminar

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
3.- El usuario cancela su decisión de borrar la colaboración.	4.- No se modifica el estado del sistema

Caso de uso: Copiar colaboración.

Actores: Representante, Administrador.

Propósito: Copiar una colaboración.

Resumen: El sistema generará una nueva colaboración con los mismos datos que la seleccionada, pero con identificador diferente.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración y pulsa la opción 'copy sponsorship'	2.- El sistema accede a los datos de la colaboración seleccionada y crea una nueva colaboración con un nuevo identificador y en estado <i>borrador</i> .

Caso de uso: Solicitar colaboración.

Actores: Representante, Administrador.

Propósito: Cambiar el estado de una colaboración a pendiente.

Resumen: El sistema, tras validar los datos de la colaboración, la pasará del estado borrador a pendiente.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración en estado borrador y hace clic sobre la opción 'submit sponsorship'	2.- El sistema muestra un mensaje confirmando la solicitud de la colaboración
3.- El usuario confirma su decisión de solicitar la colaboración.	4.- El sistema valida los datos de la colaboración. 5.- Si los datos son correctos el sistema cambia el estado de la colaboración a pendiente

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
3.- El usuario cancela su decisión de solicitar la colaboración	4.- No se modifica el estado del sistema

Curso alternativo 2

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario edita los datos del formulario y hace clic sobre la opción "submit sponsorship"	5.- Si los datos no son correctos el sistema muestra un diálogo con los errores detectados. 7.- Volvemos al punto 2 del curso típico.

Caso de uso: Aprobar colaboración.

Actores: Supervisor, Administrador.

Propósito: Aprobar una colaboración.

Resumen: El sistema, tras validar los datos de la colaboración, la pasará del estado pendiente al aprobado.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración en estado pendiente y hace clic sobre la opción 'approve sponsorship'.	2.- El sistema valida los datos de la colaboración. 3.- Si los datos son correctos el sistema cambia el estado de la colaboración a aprobada.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
4.- El usuario edita los datos del formulario y hace clic sobre la opción "approve sponsorship"	3.- Si los datos no son correctos el sistema muestra un diálogo con los errores detectados. 5.- Volvemos al punto 2 del curso típico.

Caso de uso: Denegar colaboración.

Actores: Supervisor, Administrador.

Propósito: Cambiar el estado de una colaboración a denegada.

Resumen: El sistema, tras validar los datos de la colaboración, la pasará del estado pendiente o aprobada al estado denegada.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración en estado pendiente o aprobada y hace clic sobre la opción 'reject sponsorship'.	2.- El sistema comprueba que el campo de comentarios del supervisor no este vacío. 3.- Si el campo no esta vacío el sistema cambia el estado de la colaboración a denegada.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
4.- El usuario edita los comentarios del supervisor y hace clic sobre la opción "reject sponsorship"	3.- Si el campo de comentarios del supervisor está vacío el sistema muestra un diálogo con el error detectado. 5.- Volvemos al punto 2 del curso típico.

Caso de uso: Finalizar colaboración.

Actores: Supervisor, Administrador.

Propósito: Cambiar el estado de una colaboración a finalizada.

Resumen: El sistema, tras validar los datos de la colaboración, la pasará del estado aprobada al estado finalizada.

Tipo: Primario y esencial.

Curso típico de sucesos

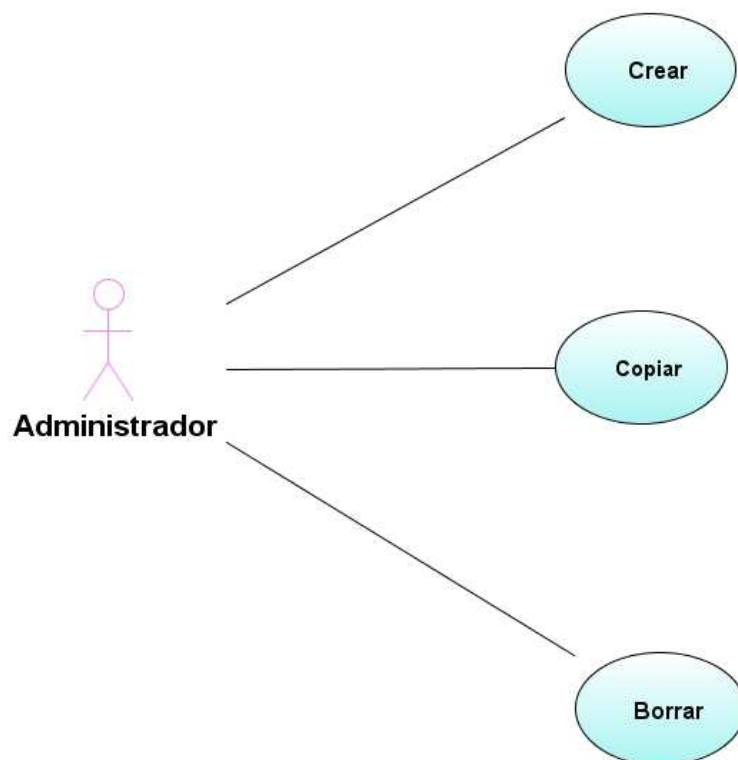
ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración en estado aprobada y hace clic sobre la opción 'finalize sponsorship'.	2.- El sistema cambia el estado de la colaboración a finalizada.

3.1.4.2 Gestión de eventos

Caso de uso: Gestión de eventos

Actores: Administrador.

Descripción: Esta funcionalidad permite gestionar todas las posibles acciones que se pueden realizar sobre una definición de evento. Encapsula un seguido de casos de uso más específicos que especificaremos a continuación. Estos casos de uso son *crear*, *copiar* y *borrar* una definición de evento.



Caso de uso: Crear definición de evento.

Actores: Administrador.

Propósito: Dar de alta una definición de evento en el sistema.

Resumen: La definición será dada de alta en el sistema con los datos que haya introducido.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona la opción 'new event definition'	
	2.- El sistema muestra el interfaz de edición de definición de eventos
3.- El usuario inserta los datos en el formulario y pulsa el botón "save".	
	4.- El sistema valida los datos introducidos.
	5.- Si los datos son correctos el sistema sale del formulario de edición.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario edita los datos del formulario y hace clic al botón "save"	5.- Si los datos no son correctos el sistema muestra un diálogo con los errores detectados. 7.- Volvemos al punto 4 del curso típico.

Caso de uso: Copiar definición de evento.

Actores: Administrador.

Propósito: Copiar una colaboración.

Resumen: El sistema generará una nueva definición de evento con los mismos datos que la seleccionada, pero con identificador diferente.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una colaboración y pulsa la opción 'copy event definition'	2.- El sistema accede a los datos de la definición seleccionada y crea una nueva definición con un nuevo identificador.

Caso de uso: Borrar definición de evento.

Actores: Administrador.

Propósito: Borrar una definición de evento.

Resumen: La definición será borrada del sistema tras confirmación del usuario.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una definición de evento y pulsa la opción 'delete event definition'	
	2.- El sistema muestra un mensaje confirmando la baja de la definición.
3.- El usuario confirma su decisión de borrar la definición.	
	4.- El sistema comprueba que la definición no este asociada con ningún evento.
	5.- Si la definición no tiene ningún evento asociado el sistema elimina toda su información.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
3.- El usuario cancela su decisión de borrar la definición.	4.- No se modifica el estado del sistema

Curso alternativo 2

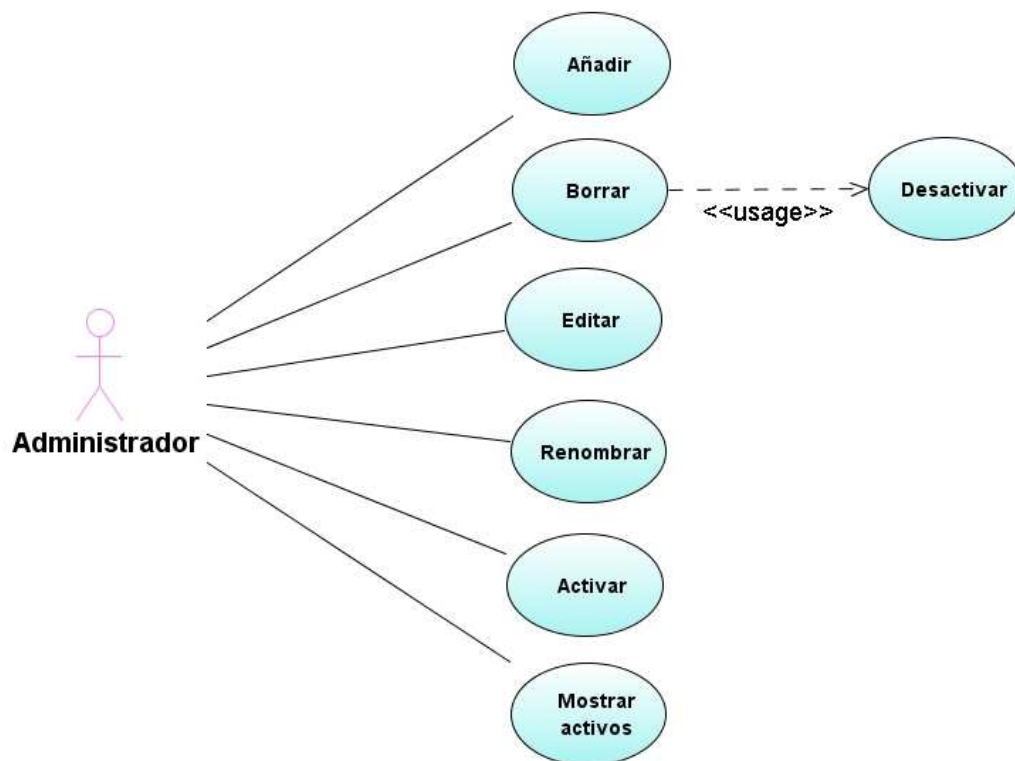
ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
	5.- Si la definición tiene algún evento asociado el sistema muestra un mensaje indicando que la definición no puede ser borrada. No se modifica el estado del sistema.

3.1.4.3 Gestión de tipologías

Caso de uso: Gestión de tipologías

Actores: Administrador.

Descripción: Esta funcionalidad permite gestionar todas las posibles acciones que se pueden realizar sobre una tipologías; por este motivo se podría decir que encapsula un seguido de casos de uso más específicos que especificaremos a continuación. Estos casos de uso son *añadir, borrar, editar, renombrar, desactivar, activar y mostrar activos*.



Caso de uso: Añadir tipología.

Actores: Administrador.

Propósito: Dar de alta una tipología en el sistema.

Resumen: La tipología será dada de alta en el sistema con los datos que haya introducido.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona un nodo del árbol de tipologías y pulsa 'add new item'	2.- El sistema añade una nueva tipología al nodo seleccionado y muestra el interfaz de edición.
3.- El usuario inserta el nombre de la tipología y pulsa 'enter'	4.- El sistema valida el nombre de la tipología.
	5.- Si el nombre es valido almacena la tipología.
6.- El usuario marca los conceptos que tendrá la tipología y pulsa 'save concepts relation'	7.- El sistema almacena los datos de la tipología.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario edita el nombre de la tipología y pulsa 'enter'	5.- Si el nombre no es válido el sistema muestra un diálogo con el error detectados. 7.- Volvemos al punto 4 del curso típico.

Caso de uso: Borrar tipología.

Actores: Administrador.

Propósito: Borrar una tipología.

Resumen: La tipología será borrada del sistema tras confirmación del usuario.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona un nodo del árbol de tipologías y pulsa 'remove selected item'	2.- El sistema muestra un mensaje confirmando la baja de la tipología.
3.- El usuario confirma su decisión de borrar la tipología.	4.- El sistema comprueba que la tipología no este asociada con ninguna colaboración.
	5.- Si la tipología no tiene ninguna colaboración asociada el sistema elimina toda su información.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
3.- El usuario cancela su decisión de borrar la tipología.	4.- No se modifica el estado del sistema

Curso alternativo 2

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario decide no desactivar la tipología	5.- Si la tipología tiene alguna colaboración asociada el sistema muestra un mensaje indicando que no puede ser eliminada, se pregunta al usuario si desea desactivarla. 7.- No se modifica el estado del sistema

Curso alternativo 3

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario decide desactivar la tipología	7.- La tipología pasa al estado desactivado

Caso de uso: Editar tipología.

Actores: Administrador.

Propósito: Editar el contenido de una tipología.

Resumen: El sistema almacenará los cambios producidos en la tipología editada.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona una hoja del árbol de tipologías.	
	2.- El sistema muestra la pantalla de edición para la tipología seleccionada
3.- El usuario marca los conceptos que tendrá la tipología y pulsa 'save concepts relation'	
	4.- El sistema almacena los datos de la tipología.

Caso de uso: Renombrar tipología.

Actores: Administrador.

Propósito: Renombrar una tipología en el sistema.

Resumen: El nombre de la tipología será modificado en el sistema.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona un nodo del árbol de tipologías y pulsa 'rename the selected item'	2.- El sistema permite la edición sobre el nodo seleccionado.
3.- El usuario inserta el nombre de la tipología y pulsa 'enter'	4.- El sistema valida el nombre de la tipología. 5.- Si el nombre es válido el sistema almacena el nuevo nombre de la tipología.

Curso alternativo 1

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
6.- El usuario edita el nombre de la tipología y pulsa 'enter'	5.- Si el nombre no es válido el sistema muestra un diálogo con el error detectados. 7.- Volvemos al punto 4 del curso típico.

Caso de uso: Activar tipología.

Actores: Administrador.

Propósito: Activa una tipología en el sistema.

Resumen: La tipología pasará a estado activa en el sistema.

Tipo: Primario y esencial.

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario selecciona un nodo del árbol de tipologías en estado desactivado y pulsa 'active the selected item'	2.- El sistema pasa a estado activo la tipología seleccionada

Caso de uso: Mostrar inactivos.

Actores: Administrador.

Propósito: Muestra las tipologías inactivas del sistema

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario pulsa 'show inactives items'	2.- El sistema muestra las tipologías que han sido desactivadas.

Caso de uso: Mostrar solo activos.

Actores: Administrador.

Propósito: Ocultar las tipologías inactivas del sistema

Curso típico de sucesos

ACCIONES DE LOS ACTORES	RESPUESTA DEL SISTEMA
1.- El usuario pulsa 'show only actives'	2.- El sistema oculta las tipologías desactivadas.

3.2 Modelo conceptual

Tras el proceso de toma de requerimientos se desarrollaremos el modelo conceptual. Por tal de facilitar su comprensión lo desglosaremos conforme lo vayamos enunciando.

3.2.1 Colaboración

Una *colaboración* es un elemento consistente en una conjunto de beneficios a los que llamaremos *conceptos*. De una colaboración nos interesa conocer el identificador, el año al que pertenece, el departamento y cuenta asociado, la fecha de creación y comentarios.

El identificador de la colaboración consiste en el id del usuario que crea la colaboración seguido de 4 dígitos.

3.2.2 Evento y definición de evento

Las colaboraciones se pueden agrupar en *eventos*, de un evento nos interesa saber su identificador y la fecha de creación.

Cada evento debe tener una definición asociada. Para cada definición almacenaremos su identificador, año, descripción, departamento, cuenta, guía y si está activo.

3.2.3 Costes

Una definición está formada por un coste directo y otro indirecto. Una colaboración tiene un coste directo.

Los costes representan al conjunto de conceptos que forman la colaboración.

Una colaboración contenida en un evento heredará el coste directo asignado en la definición de dicho evento.

3.2.4 Conceptos

Un concepto es un beneficio hacia una persona o entidad que representa un coste para la empresa. Los conceptos se dividen en directos o indirectos.

La lista de conceptos directos es: 'Inscription', 'Miscellanius', 'Other external services', 'Other', 'Travel', 'Accomodation' y 'Goods'.

La lista de conceptos indirectos es: 'Goods', 'External services', 'Infrastructure' y 'Other'

Los conceptos tienen como atributos su descripción y el orden en el que aparecerán, además los conceptos directos tendrán asociada una determinada imagen.

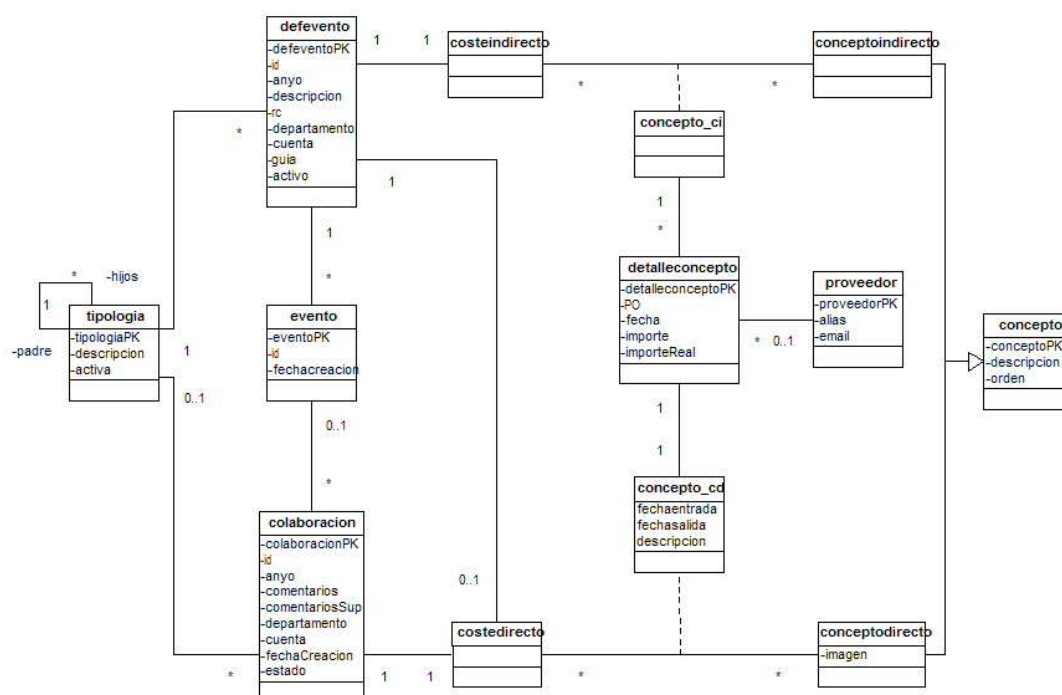
3.2.5 Coste - Concepto

Los costes directos agrupan los conceptos directos de la colaboración. Los costes indirectos agrupan a los conceptos indirectos.

Para cada asociación entre un coste y un concepto almacenaremos sus detalles, estos detalles están compuestos por la PO (purchased order), el importe estimado y el real. Además podrá tener asignado un proveedor. De un proveedor necesitaremos saber su alias y su email.

Para la asociación de conceptos y costes indirectos el sistema deberá almacenar un histórico de sus detalles.

Para la asociación entre conceptos y costes directos el sistema deberá almacenar una descripción, y para determinados conceptos, almacenará también fechas de inicio y fin.

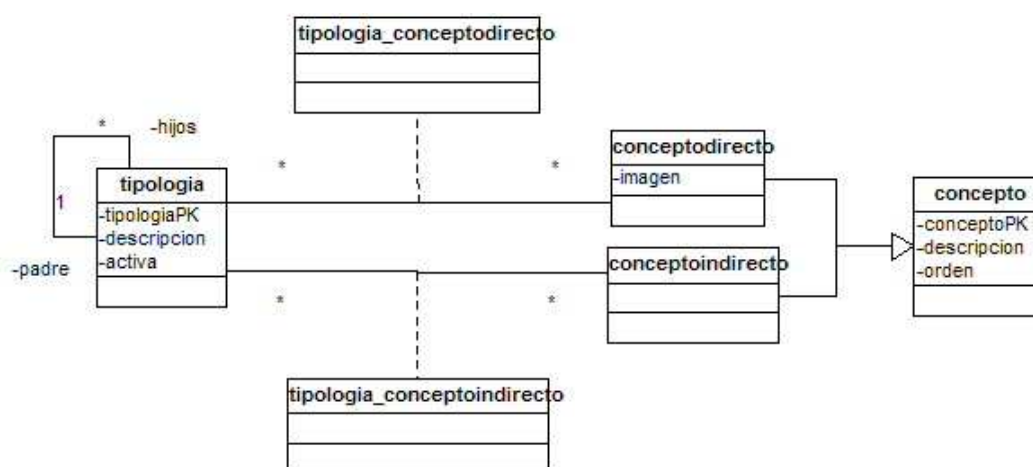


3.2.6 Tipología

La tipología a la que pertenece una colaboración determina el grupo de conceptos que se marcarán por defecto. La selección de tipologías está organizada como un árbol, es decir una tipología tiene un padre y puede tener hijos. Las tipologías que no tienen hijos son seleccionables en la colaboración.

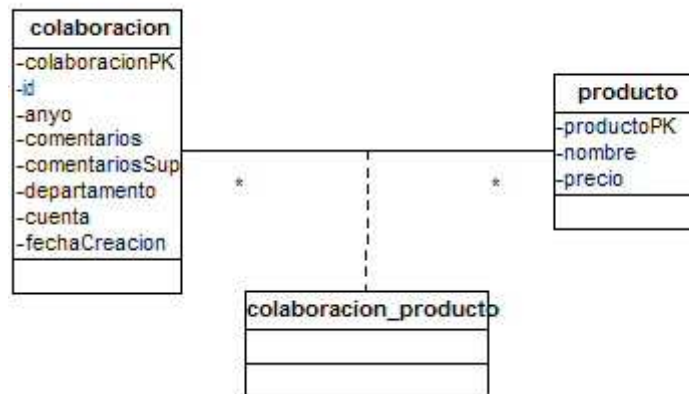
Las colaboraciones que no están asociadas a un evento y las definiciones de eventos tienen asignada una tipología.

Las colaboraciones asociadas a un evento heredan la tipología de la definición de dicho evento, y por consiguiente los conceptos definidos por la tipología.



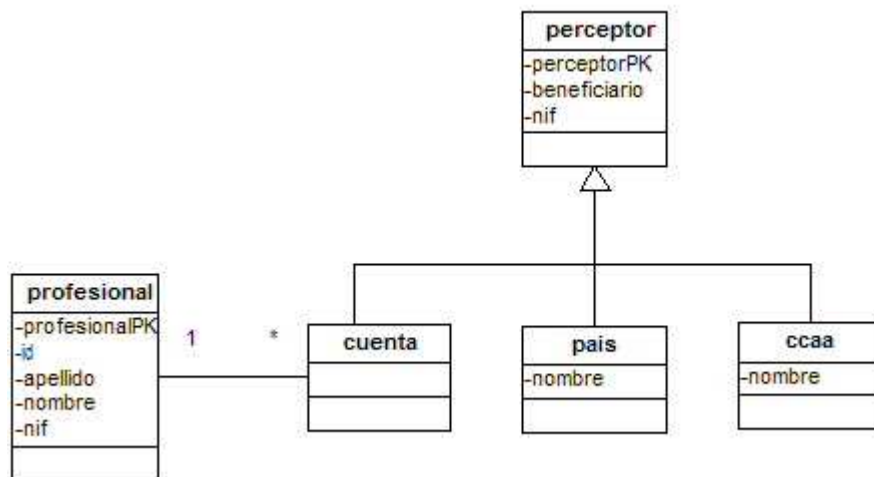
3.2.7 Producto

Las colaboraciones se crean sobre alguno de los productos de la empresa. De los productos necesitamos conocer su nombre y precio.



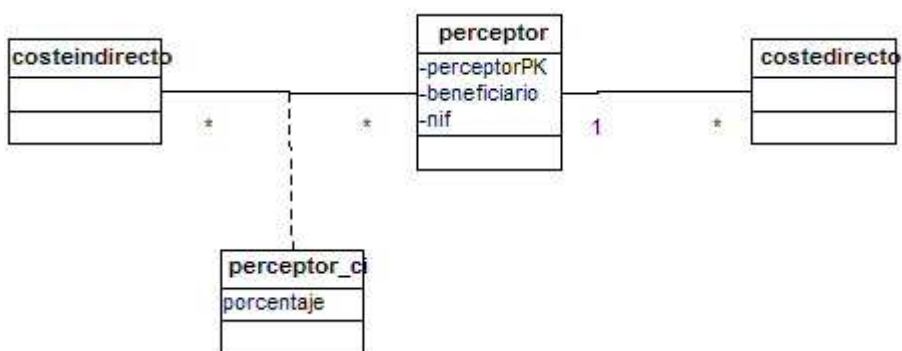
3.2.8 Perceptor

El perceptor de los beneficios o conceptos de la colaboración puede ser una cuenta, un país o una comunidad autónoma. Del perceptor necesitaremos saber el nombre (beneficiario) y su NIF. En caso de tratarse de un país o comunidad autónoma almacenaremos su nombre. En caso que el perceptor sea de tipo cuenta almacenaremos un profesional perteneciente a alguno de los centros médicos con los que la empresa trata. De los profesionales necesitamos saber su identificador, apellido, nombre y nif.



3.2.9 Perceptor - Coste

Los perceptores se asocian con las colaboraciones a través de los costes, recordemos que los costes representan al conjunto de conceptos de la colaboración. Los costes directos tienen un solo perceptor. Los indirectos pueden tener más de uno, en cuyo caso almacenaremos el porcentaje asignado a cada uno de ellos. La suma de dichos porcentajes deberá ser siempre cien. Todos los perceptores de un coste serán del mismo tipo.

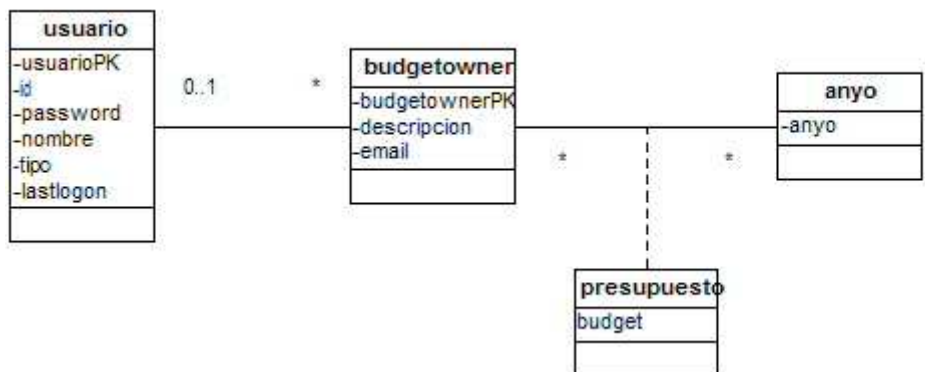


3.2.10 Budget Owner

Tal y como hemos comentado las colaboraciones tiene una serie de conceptos marcados, estos conceptos representan un gasto para la empresa.

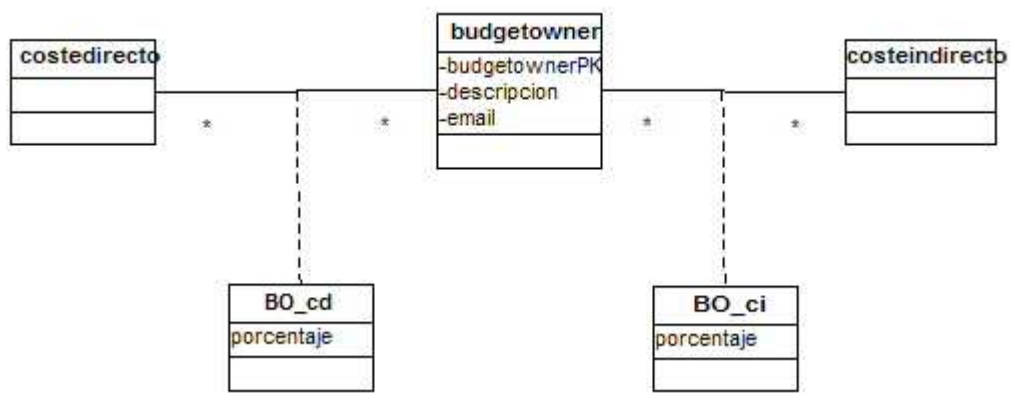
La empresa dispone de una serie de presupuestos anuales para hacer frente a estos gastos. Los propietarios de estos presupuestos (Budget Owners) disponen de un presupuesto anual. Estos Budget Owners se pueden corresponder con un usuario del sistema.

De los Budget Owners necesitamos conocer su nombre (descripción) y su email. Para los usuarios almacenaremos su identificador, password, nombre, tipo y fecha de la última entrada al sistema.



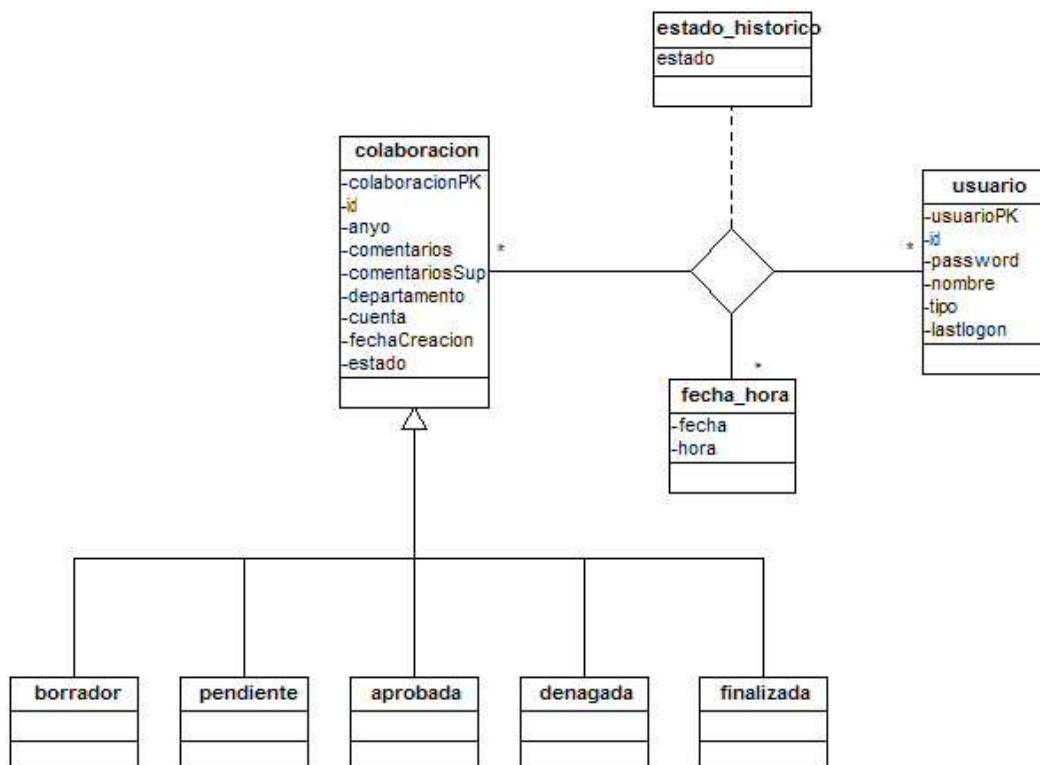
3.2.11 Budget Owner - Coste

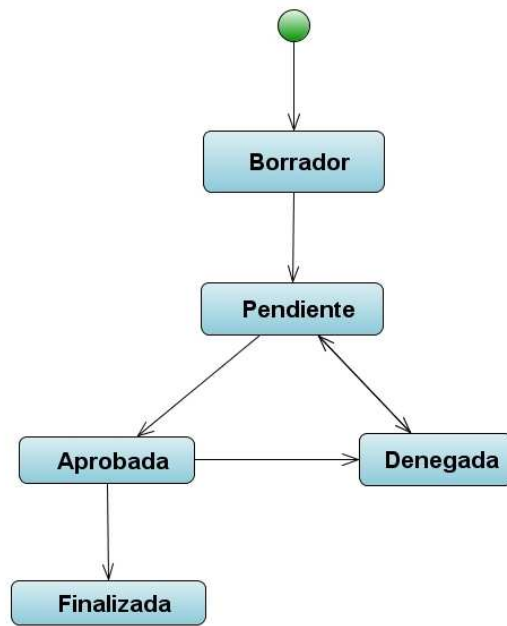
Al igual que ocurría con los perceptores, los Budget Owners también se asocian a las colaboraciones a través de sus costes. Los costes directos e indirectos tendrán uno o más Budget Owners asociados, deberemos almacenar el porcentaje asignado a cada uno de ellos. La suma de dichos porcentajes será siempre cien.



3.2.12 Estados

Una colaboración pasa por distintos estados, estos son: borrador, pendiente, aprobada, denegada o finalizada. Una colaboración solo puede estar en alguno de estos estados a la vez. Son los usuarios los que hacen cambiar a la colaboración de estado. El sistema deberá mantener un histórico de *quien y cuando* se cambió a *que estado* una colaboración.





4 ● Implementación y pruebas

A partir de las especificaciones obtenidas en el capítulo anterior procederemos al diseño y posterior implementación del sistema. Para ello deberemos tener en cuenta aspectos como la tecnología de la que disponemos.

Contenidos de la sección

4.1 Diseño

4.2 Implementación

4.3 Pruebas

4.1 Diseño

Antes de pasar a la implementación deberemos completar la etapa de diseño. Esto es debido a que el resultado de la ejecución de la etapa de la especificación, no es suficiente para poder comenzar a ejecutar la etapa de implementación del sistema. Esta condición de insuficiencia es debida a que el output de la etapa de especificación está orientada hacia la descripción de qué debe hacer el sistema pero no está enfocada a resolver la cuestión de cómo debe hacerlo.

Esta etapa recibirá como input el resultado procedente de la ejecución de la etapa de especificación, y producirá un output, que permitirá comenzar a ejecutar la etapa de implementación del sistema.

El objetivo principal de éste capítulo es exponer y justificar las decisiones que se han tomado a la hora de desarrollar nuestro sistema.

Debemos seleccionar el diseño arquitectónico a elegir, dicho diseño deberá satisfacer requerimientos como la mantenibilidad, fiabilidad, o eficiencia.

Uno de los requisitos del sistema especifica que el acceso deberá realizarse accediendo a un servidor web a través de Internet mediante un navegador. En la ingeniería del software esto se denomina aplicación web.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que soportan por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

4.1.1 Arquitectura cliente-servidor

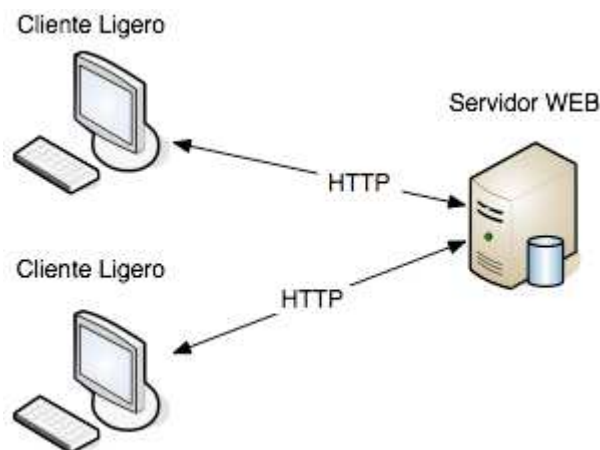
Las aplicaciones web requieren de una arquitectura cliente-servidor. Esta arquitectura consiste básicamente en que un programa (el cliente) que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más

importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.



4.1.1.1 Características de un cliente

En la arquitectura cliente-servidor el remitente de una solicitud es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

4.1.1.2 Características de un servidor

En los sistemas cliente-servidor el receptor de la solicitud enviada por cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

4.1.1.3 Comparativa con otras arquitecturas

A continuación compararemos la arquitectura cliente servidor con otras arquitecturas conocidas.

- Redes de pares: Las redes de pares, también conocidas como redes par-a-par o peer-to-peer (abreviado con las siglas P2P) son otro tipo de arquitectura de red; en ellas cada nodo o elemento del sistema puede actuar al mismo tiempo como cliente y como servidor; cada nodo tiene, por tanto, las responsabilidades y estados de ambos elementos.

- Cliente-Cola-Cliente: Si bien la clásica arquitectura cliente-servidor requiere uno de los puntos terminales de comunicación para actuar como un servidor, que puede ser algo más difícil de aplicar, la arquitectura Cliente-Cola-Cliente habilita a todos los nodos para actuar como clientes simples, mientras que el servidor actúa como una cola que va capturando las peticiones de los clientes (un proceso que debe pasar sus peticiones a otro, lo hace a través de una cola, por ejemplo, una consulta a una base de datos, entonces, el segundo proceso conecta con la base de datos, elabora la petición, la pasa a la base de datos, etc.). Esta arquitectura permite simplificar en gran medida la implementación de software. la arquitectura P2P originalmente se basó en el concepto "Cliente-Cola-Cliente".

Ventajas

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de cliente-servidor que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

Desventajas

- La congestión del tráfico ha sido siempre un problema en el paradigma de cliente-servidor. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuanto más nodos hay, mejor es el ancho de banda que se tiene.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.

4.1.2 Arquitectura de Software

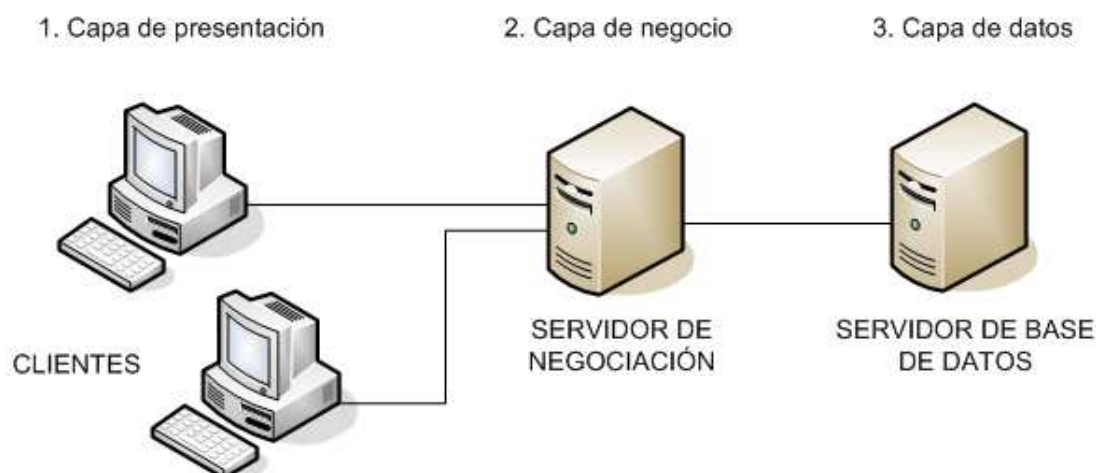
La Arquitectura de Software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

Para la construcción de nuestro sistema utilizaremos una arquitectura en tres capas.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

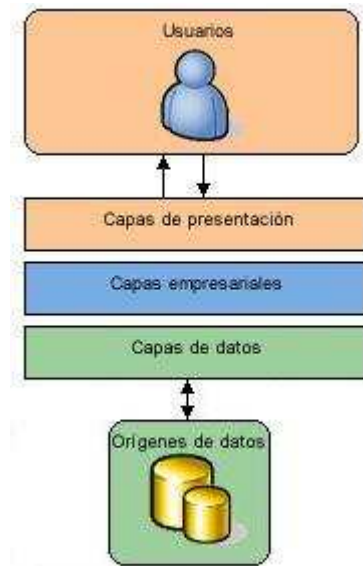


4.1.2.1 Capas y niveles

Capa de presentación: es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz grafica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



4.1.2.2 Justificación

- El diseño de una aplicación que utilice una arquitectura de capas presenta la gran ventaja de que producirá código modular en el que la modificación de uno de sus componentes no exige modificar otro de los componentes pertenecientes a otra capa.
- El diseño en capas, también favorece la posible reutilización de código para futuras aplicaciones.

4.1.3 Metodología de diseño

La implementación de nuestro sistema la realizaremos a través de un lenguaje orientado a objetos.

La idea principal de la orientación a objetos es subir el nivel de programación al nivel de objetos, ya que un objeto es algo mucho más intuitivo y comprensible por el ser humano, que los términos que se utilizan normalmente en programación como bytes, bits, registros etc...

Un objeto es un paquete de datos del sistema que tiene una serie de atributos y de métodos, que pueden utilizar los datos que contiene el objeto.

Los atributos son las operaciones más relevantes del objeto y los métodos son los instrumentos necesarios para manipular estos atributos.

Las clases de objetos definen como son los objetos, es decir, los tipos de datos que guardarán y los métodos que tendrán.

La programación orientada a objetos, estructura un programa como un conjunto de objetos, que se pueden comunicar entre ellos para realizar tareas.

4.1.3.1 Justificación

- La orientación a objetos ayuda a hacer programas que tengan módulos más fáciles de construir, de mantener y de reutilizar.
- Ocultación de la información.
- Abstracción.
- Herencia.
- Polimorfismo.
- El coste de la implementación es menor que, aplicando otro tipo de patrones.
- Permite dividir las responsabilidades más fácilmente dentro de un mismo grupo de trabajo.
- Los sistemas que utilizan el patrón de orientación a objetos resultan más viables económicamente a medio y a largo plazo.
- Aumenta la propiedad de extensibilidad, gracias a la herencia y al permitir añadir nuevos módulos.

4.1.4 Estudio tecnológico

En este apartado realizaremos una comparativa entre diferentes tecnologías con el objetivo de encontrar las que se adecuen a nuestro sistema

A la hora de tomar las decisiones tecnológicas para este proyecto, se han debido tener en cuenta los siguientes factores:

Factor Aplicación:

Debemos encontrar las tecnologías que mejor se adapten a nuestro sistema, teniendo en cuenta que nuestra aplicación es un módulo que se deberá integrar en una aplicación web.

Factor Empresa:

También hay que tener en cuenta que algunas decisiones a la hora elegir tecnología están limitadas por imposiciones de la empresa.

4.1.4.1 Tecnologías y herramientas utilizadas

En este apartado iremos desglosando las decisiones tomadas en la elección de las tecnologías para la *plataforma de programación y servidor web*.

4.1.4.1.1 Plataforma de programación

En este apartado se decidirá el entorno de computación que sea capaz de ejecutar aplicaciones desarrolladas usando un determinado lenguaje de programación.

En este punto partimos de dos opciones sólidas:

- JEE
- Framework .NET

Ha continuación realizaremos un estudio comparativo entre ambas plataformas.

Presentación

JEE es la edición empresarial de la plataforma Java. Aprovecha las fortalezas de la edición estándar de Java (J2EE), complementándolas con especificaciones, funcionalidades y lineamientos orientados al desarrollo de aplicaciones empresariales. El nombre original de JEE era Java 2 Enterprise Edition (J2EE) (y es como la mayoría del mundo lo sigue llamando), sin embargo, a partir de la edición 5, que es la más reciente, se cambió el nombre

oficial a Java EE, o JEE, así que, esa es la nomenclatura que utilizaremos.

El framework .NET es una plataforma creada por Microsoft, para el desarrollo y ejecución de aplicaciones modernas. La primera versión de .NET se lanzó en el 2002, en gran parte como respuesta al éxito de J2EE. Actualmente, .NET se encuentra en su versión 2.0.

Ejecución de aplicaciones

JEE: El código fuente de Java se compila en un byte-code, que es independiente de sistema operativo y arquitectura de procesador. Dicho bytecode se ejecuta por una máquina virtual de Java (JVM). En las primeras versiones de Java, los JVM sólo eran capaces de interpretar código y generar las llamadas correspondientes al procesador (en lugar de generar código nativo), afectando seriamente su desempeño. Sin embargo, todos los JVM modernos tienen la capacidad de hacer compilación Just-In-Time (JIT) para generar código nativo, con lo cual se obtiene un mejor desempeño y se puede aprovechar las características especiales de cada arquitectura de procesador.

.NET: De forma similar que en JEE, en .NET, el código fuente se traduce a un lenguaje intermedio, llamado MSIL (Microsoft Intermediate Language). Esta tecnología proporciona servicios como verificación de código, manejo de memoria, colección de basura y seguridad. El MSIL se ejecuta por un componente denominado CLR (Common Language Runtime), el cual hace compilación JIT. El código traducido es cacheado de tal forma, que la siguiente vez puede ejecutarse directamente.

JEE

JEE está basado en una arquitectura del lado del servidor (Served-based). Este tipo de arquitectura, concentra la mayoría de los procesos de la aplicación en el servidor, o en un pedazo de éste. Otorgando dos ventajas críticas:

- **Múltiples Clientes:** una arquitectura basada en el servidor, requiere una clara separación entre la capa cliente (interfaz) y la capa servidor, en la cual se realizan los procesos de la aplicación. Esto permite que una simple aplicación soporte simultáneamente, clientes con distintos tipos de interfaces.

- **Operaciones robustas:** una arquitectura basada en el servidor soporta escalabilidad, confiabilidad, disponibilidad y recuperabilidad. Aplicaciones basadas en el servidor, pueden ser divididas y distribuidas en múltiples procesadores. Componentes de la aplicación, pueden replicarse, para dar soporte a caídas instantáneamente.

La plataforma JEE se define por medio de una especificación, desarrollada a través del Java Community Process. Aunque no es un estándar formal, avalado por una organización como ISO o ECMA, JEE es informalmente considerado un estándar, ya que cualquier proveedor puede desarrollar servidores de aplicación, y ambientes de ejecución para cualquier plataforma, siempre y cuando cumplan los requerimientos establecidos en la especificación, los cuales son verificados a través de un conjunto de pruebas.

El hecho de que Java corra sobre un ambiente de ejecución intermedio, como lo es el JVM, más la disponibilidad de JVMs para prácticamente cualquier plataforma, le da a Java su capacidad multiplataforma, expresada en la frase, Write Once, Run Anywhere (escribe una vez, ejecuta en cualquier lado).

.NET

Está diseñado principalmente para el sistema operativo Windows. Lo que tiene ciertas ventajas, por ejemplo, el manejo de ventanas para aplicaciones de escritorio es muy bueno. Sin embargo, el principal inconveniente, está en la portabilidad de las aplicaciones.

Una alternativa para mitigar esta limitante, es recurrir a Mono, una implementación multiplataforma y open source del ambiente de ejecución de .NET. Pero, hay que tener en cuenta que Mono no implementa todos los detalles de la plataforma .NET, así que si no estamos al pendiente, podemos terminar con aplicaciones que funcionen sobre el CLR de Microsoft, pero no sobre Mono.

Lenguajes

JEE

En este caso, el único lenguaje soportado es Java, y es el que se debe utilizar para el desarrollo de todos los componentes. Existen sólo dos formas oficiales para acceder a la plataforma JEE con otros lenguajes, la primera es a través de JNI (Java Native Interface) y la segunda, es a través de la interoperabilidad que ofrece CORBA.

.NET

La plataforma .NET es una plataforma de desarrollo de software, con especial énfasis en la independencia de lenguaje, y la transparencia a través de redes. Tal vez los dos lenguajes más representativos de esta plataforma son: C#, y Visual Basic .NET. No obstante, también es posible generar aplicaciones para .NET en Cobol, C++, y Fortran, entre otros. La filosofía es permitir que los desarrolladores utilicen el lenguaje(s) de su preferencia, pero aprovechando los servicios y beneficios de la plataforma.

Capacidades Empresariales

Las tareas más comunes en cómputo empresarial son: transacciones, distribución de objetos remotos y soporte de servicios Web XML. Veamos cómo las manejan ambas plataformas.

Capacidades empresariales: Transacciones

El objetivo de las transacciones, es mantener los datos del sistema coherentes, cuando las cosas no marchan bien. Típicamente recurrimos al acrónimo ACID, para recordar las propiedades que debe tener una transacción: atomicidad, consistencia, aislamiento y durabilidad. Por ejemplo, cuando un cajero automático te entrega una cantidad en efectivo, al mismo tiempo la deduce de la cuenta. Ningún tipo de fallo o imprevisto, debería provocar que el cajero realice una operación, sin realizar la otra. Es decir, todas las partes de la transacción deben aplicarse, o ninguna; y sin importar lo que pase, el sistema se deja en un estado consistente.

JEE

Los desarrolladores pueden codificar la gestión de transacciones de forma explícita (manual), o especificar el desempeño requerido y dejar que el contenedor se encargue (modo automático). En la mayoría de los casos, los desarrolladores intentan delegar el manejo transaccional al contenedor EJB. Manejar manualmente las transacciones, puede introducir errores sutiles en la aplicación; de hecho, el contenedor ha probado ser más eficiente que los desarrolladores.

.NET

El CLR soporta tanto transacciones manuales, como automáticas. Con las primeras, los desarrolladores inician la transacción, enlistan otros en la transacción, aplican o abortan, y finalizan la transacción. Con las segundas, los desarrolladores definen un comportamiento transaccional de objeto, ajustando un valor de atributo transaccional en la página ASP.NET, un método de servicio Web XML, o una clase. Cuando un objeto se marca para participar en una transacción, automáticamente se ejecutará dentro del alcance de la transacción.

Capacidades empresariales: objetos remotos

Tanto en .NET como JEE, la lógica de negocios puede encapsularse en componentes que existan de manera local (en memoria), o fuera del proceso; en una máquina remota. Como sea, ambas arquitecturas siguen diferentes guías sobre cómo lograr la distribución.

JEE

La transparencia de localidades es un atributo base de JEE. JNDI encuentra componentes del lado del servidor, tales como JMS Queues o EJBs, los cuales, si se ejecutan en un ambiente con clusters, pueden o no, residir en la misma máquina virtual. Por lo general, todos los accesos a un EJB deben ir a través de interfaces, que permiten al servidor de aplicaciones, implementar su propio balanceo de carga, a través de múltiples nodos para optimizar los tiempos de respuesta y el tráfico. En términos prácticos, la mayoría de los servidores de aplicación, intentan comunicar ejes en la misma VM para minimizar el tráfico de red, y la sobrecarga asociada. Los desarrolladores pueden forzar la ejecución en una misma VM con interfaces locales.

.NET

Con .NET remoting, los desarrolladores pueden llamar objetos remotos distribuidos, a través de dominios de aplicación, procesos y agrupaciones de máquinas. .NET remoting oculta los desagradables detalles del llamado remoto, formatea los mensajes y provee los mecanismos de transporte entre capas. El punto importante es, cómo decidir cuándo usar o no remotos. En .NET no se decide automáticamente distribuir los objetos como en JEE, debido al precio de permanencia en la red. Claro, se debe entender qué se recibe a cambio. En el mundo .NET, los desarrolladores emplean remotos para incrementar la seguridad y facilitar el mantenimiento, pero no para incrementar la escalabilidad; eso se logra de manera simple: ampliando el número de servidores Web.

Capacidades empresariales: Servicios Web XML

JEE

Provee soporte para la creación de servicios Web, donde cada EJB de servicio Web puede ser consumidor, y un bean de sesión sin estado, puede proveer un servicio Web. Además enfatizan el soporte de servicios Web de la plataforma Java, incorporando respectivamente funciones de nomenclatura (naming), búsqueda (lookup), invocación (invocation) y uso (usage).

NET

Microsoft colocó los servicios Web como parte del núcleo de .NET. Este contiene los estándares más recientes de servicios XML aceptados, tales como el estándar XML Schema.

Otras variables a considerar

A continuación, enlistamos las variables que la mayoría de los expertos proponen como vitales, para tomar una decisión de esta índole.

Otras variables a considerar: Escalabilidad

La infraestructura tecnológica debe ser capaz de brindar servicio a plataformas heterogéneas, provenientes de nuevos clientes, o de fusiones a nivel empresarial. Si se prevé un alto nivel de heterogeneidad, en las "relaciones tecnológicas" de la empresa, sin duda JEE será la opción más inteligente. De otra forma, la elección queda sujeta a otras variantes.

Otras variables a considerar: Integración

La línea del tiempo, también debe verse en perspectiva con el pasado de la empresa. La integración de sistemas antiguos y de tecnologías en desuso, con los nuevos esquemas, debe planificarse cuidadosamente. Tanto JEE como .NET, proveen herramientas específicas de integración de sistemas legados, dirigidas a segmentos de mercado específicos.

Otras variables a considerar: Independencia

Hay empresas que consideran vital, no depender exclusivamente de un proveedor final de tecnología, sino beneficiarse de la competencia que existe entre diversas empresas proveedoras. Por otro lado, hay quienes se sienten más seguros al trabajar con un conjunto de herramientas coherentes entre sí, y con un robusto esquema de soporte, debido a que provienen de un sólo proveedor.

Otras variables a considerar: Costes de desarrollo

Un cambio o introducción de tecnología, sin una profunda justificación acorde a las políticas del negocio, es decir, dictada por las tendencias del mercado u otros criterios superficiales, puede resultar, en primera instancia, mucho más caro de lo que aparenta. Se deben considerar aspectos de fondo tales como, las capacidades del personal actual, los costos de entrenamiento y transferencia del conocimiento, el soporte existente en el medio, etcétera. Aún así, uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún coste. La implementación Java EE de Sun Microsystems puede ser descargada gratuitamente, y hay muchas herramientas de código abierto disponibles para extender la plataforma o para simplificar el desarrollo.

Otras variables a considerar: Robustez

Debemos considerar el tipo de desarrollos que necesita, soporta o provee nuestra empresa. Podemos optar por las herramientas de desarrollo rápido, que una u otra tecnología ofrecen; o probablemente, preferimos una estrategia probada y con un amplio universo de empresas que la soporten. Esto depende de parámetros tales como, la criticidad de los sistemas, y el tiempo deseado de entrega. También es importante decir que, empleando las herramientas apropiadas, un desarrollo robusto no necesariamente requiere un ciclo de producción largo.

Otras variables a considerar: tecnologías emergentes

Otro punto a considerar, puede ser el de innovación vs. estabilidad. En este sentido, .NET tiende a ser más innovador; incorporando y proponiendo nuevas capacidades. Por otro lado, JEE es más conservador en su adopción de tecnologías, esperando que algo se pruebe bien, y no haya

duda de que se mantendrá en el mercado por un buen tiempo, para entonces adoptarlo. Un ejemplo claro de esto, es el soporte a Web Services, donde .NET tomó la batuta, aun corriendo el riesgo de soportar tecnologías antes de que estuvieran estandarizadas; mientras que JEE esperó, antes de entrarle de lleno.

Decisión

JEE es más maduro, y por lo tanto, es más común encontrarlo en aplicaciones de misión crítica. Por otro lado, al ser más reciente .NET ha aprendido tanto de los aciertos como de los errores de JEE y esto le da cierta ventaja.

En cuanto a la portabilidad, aún domina JEE. Aunque el hecho de que sea un producto sin respaldo de Microsoft y que dependa de una comunidad de código abierto, afectan la confiabilidad.

Un aspecto sin discusión, es la diversidad de los lenguajes de programación. .NET ofrece soporte para un extenso número de lenguajes, mientras que JEE únicamente soporta Java.

Para una compañía que cuenta con programadores que dominan distintos lenguajes, la elección más cómoda y común, será la plataforma que ofrece soporte para múltiples lenguajes.

El hecho de que el lenguaje Java sea hoy en día el más usado en el mundo, puede quizá, atenuar esta diferencia.

Tras valorar las diferencias entre ambas plataformas, acabó primando la escalabilidad y homogenización que ofrece JEE. El hecho que el cliente ya hubiera trabajado con éxito anteriormente con dicha plataforma, así como la intención de crear un producto escalable y de un coste de producción

moderado, acabó decantando la decisión final de optar por JEE.

4.1.4.1.2 Servidor Web

Un servidor Web es un programa que implementa el *protocolo HTTP* (*hypertext transfer protocol*). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir páginas Web o páginas HTML (*hypertext markup language*).

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una *página web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Un servidor Web se mantiene a la espera de *peticiones HTTP* por parte de un *cliente HTTP* que solemos conocer como *navegador*. El cliente realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

El servidor Web que se elija debe soportar la plataforma de programación elegida para la implementación. Teniendo en cuenta que la plataforma escogida es JEE, las opciones más sólidas son:

- Apache Tomcat
- GlassFish
- Jboss

Esta primera elección se realiza teniendo en cuenta que buscamos un producto cuyo coste se adecue a los requerimientos económicos de la empresa.

Todos los servidores web listados son gratuitos, lo que reducirá considerablemente el presupuesto presentado a la empresa.

Presentación

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

GlassFish es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito y de código libre, se distribuye bajo la licencia CDDL y la GNU GPL. GlassFish tiene como base al servidor *Sun Java System Application Server* de Sun Microsystems.

Jboss es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios con una licencia de código abierto, JBoss puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia.

Comparativa

A continuación enumeramos las características principales de un servidor Web Java.

Java EE compliance

Se trata de una característica importante si la aplicación Java debe ser plenamente compatible con un servidor Sun Java EE.

EJB (Enterprise JavaBeans)

Enterprise JavaBeans (EJB) es una arquitectura de componentes Java para el desarrollo de aplicaciones en Java EE servers. Le permite empaquetar funcionalidades en componentes reutilizables. Las versiones anteriores de EJB se consideran difíciles para el desarrollo, a pesar de que estaban destinadas a ser una ventaja de desarrollo.

JSP and Servlet

Se trata de la característica más básica para cualquier servidor java.

JSF support

JavaServer Faces se trata de una arquitectura de desarrollo Web de aplicaciones Java EE. Esta arquitectura hace que el desarrollo de las interfaces de usuario sean más sencillas

Custom plug-in support

Consiste en una arquitectura para la incorporación de nuevas funciones y características en el servidor

Business-rules engine support

La mayoría de las aplicaciones se articula en torno a algún conjunto de reglas de negocio, también llamada lógica de negocio. Un componente Business-rules engine puede hacer que la programación de la lógica del negocio sea mucho más fácil de gestionar.

Hibernate support

Es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Clustering support

Consiste en ejecutar varios servidores en paralelo, que proporcionan los mismos servicios, dando lo que suele denominarse alta disponibilidad: más clientes pueden ser soportados por su cluster de servidores. Si uno de los servidores en un clúster cae, los demás servidores seguirán siendo de apoyo a los clientes. Las aplicaciones Java se mantienen accesibles, aun cuando un servidor en el grupo muere. La agrupación proporciona escalabilidad para aplicaciones empresariales Java y mejora el rendimiento.

Netbeans IDE support

Teniendo en cuenta que la herramienta de desarrollo utilizada es Netbeans, esta característica hace referencia a la capacidad de ser integrado en dicha herramienta.

Tras comparar las características de los servidores Web, podemos resumirlas en el siguiente cuadro.

	Tomcat	GlassFish	Jboss
Compilador Java EE 5	★	★	★
JSP and Servlet	★	★	★
EJB capable	★	★	★
JavaServer Faces support	★	★	★
Custom plug-in support		★	★
Business-rules engine support	★	★	★
Hibernate support	★	★	★
Clustering support	★	★	★
Netbeans IDE connector support	★	★	
Rating	8/9	9/9	8/9

Decisión

Cualquiera de los tres servidores de aplicaciones Java que hemos detallado serian totalmente válidos para soportar nuestra aplicación.

El peso de la decisión queda entonces supeditado a uno de los requerimientos no funcionales (2.4.8) que dice así:

“

El sistema debe funcionar sobre un servidor de aplicaciones Tomcat. Este servidor estará ubicado en las dependencias de la compañía y no será de uso exclusivo de nuestro sistema. Esto implica que otras aplicaciones estarán en funcionamiento en el mismo servidor que el nuestro.

4.2 Implementación

Tras haber concluido las etapas de especificación y diseño, ya estamos preparados para ejecutar la implementación.

Las decisiones tecnológicas básicas han sido ya resueltas en el apartado de diseño, con lo que para la etapa de implementación ya disponemos de todo lo necesario para empezar.

4.2.1 Vistas del sistema

En este apartado se mostrarán algunas de las vistas o interfaces gráficas de la aplicación.

¿Qué es una interfaz gráfica?

Interfaz gráfica de usuario o GUI (GUI es un acrónimo del vocablo inglés *Graphical User Interface*) es un método para facilitar la interacción del usuario con el sistema software a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas) además de texto.

La interfaz gráfica de usuario en informática, es un tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones, mediante representaciones visuales, o sea, es un instrumento que permite comunicar al usuario con un sistema software.

El usuario puede comunicarse con el sistema software, gracias a la interfaz gráfica, mediante el ratón, el teclado, una pantalla táctil y la voz entre otros métodos. En conclusión que el usuario se puede comunicar a través de la interfaz gráfica con el sistema software, mediante el uso de periféricos.

Cuando el usuario se comunica con la interfaz gráfica usando algún periférico, el sistema software detecta los eventos del periférico, y en respuesta puede hacer que la capa de presentación muestre información al usuario o bien recoja información del usuario.

Una interfaz gráfica ha de ser amigable, usable e intuitiva. La interfaz ha de tener en cuenta los usuarios finales, para los que está destinado el sistema software a la cual representa, y su entorno social y cultural.

El requisito no funcional 2.4.3 hace referencia a la interfaz de usuario y dice así:

“

La interfaz de usuario será lo más similar posible a las aplicaciones clásicas de escritorio. Esto implica minimizar al máximo las recargas del navegador, así como incluir en las interfaces componentes tales como menús o barras de herramientas

Para cumplir este requisito disponemos de la tecnología AJAX.

Tecnología AJAX

La tecnología **AJAX** (acrónimo para Asynchronous JavaScript And XML) hace uso de JavaScript asíncrono más XML para el desarrollo de aplicaciones Web interactivas.

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Framework AJAX

Un framework asiste al trabajo de un programador AJAX en dos niveles: en el lado del cliente, ofreciendo funciones JavaScript para enviar peticiones al servidor. En el lado del servidor, el cual procesa las peticiones, busca información, y la transmite al navegador. Algunos framework son muy robustos y proveen una librería completa para construir aplicaciones web.

Los Frameworks JavaScript se ejecutan en el navegador (browser-side) y son ampliamente usados en el desarrollo de aplicaciones AJAX. Existen cientos de frameworks en el mercado los más usados son:

- **Dojo toolkit**, Toolkit Modular JavaScript.
- **ExtJS**, librería que expande Prototype, JQuery y YUI.
- **jQuery**, provee un framework Ajax y muchas otras utilidades.
- **Mootools**, un framework compacto y modular mejor conocido por sus transiciones y efectos.
- **Prototype**, provee framework Ajax y muchas otras utilidades.
- **Script.aculo.us**, es utilizado con Prototype principalmente para animaciones y desarrollo de interfaces.
- **Yahoo! UI Library**, un conjunto de utilidades y controles para construir aplicaciones web enriquecidas usando técnicas como DOM scripting y AJAX.

Los frameworks citados tienen características similares, por lo que se decide valorar tres características: velocidad, estética y documentación.

Finalmente se opta por **ExtJS** ya que cumple con creces estas características.

Dispone de una amplia documentación, la velocidad es aceptable y dispone de diversos estilos, de estética agradable, que tienen el visto bueno del cliente.

Procedemos a mostrar las interfaces gráficas que conforman nuestro sistema.

4.2.1.1 Menús de navegación

Se decide realizar la navegación por pestañas, estas pestañas se colocan arriba de la interfaz con el objetivo de dejar más visibilidad en los laterales, permitiendo así una estructura de los formularios más espaciada y clara.



Cada pestaña puede a su vez contener pestañas auxiliares, este es el caso de administración

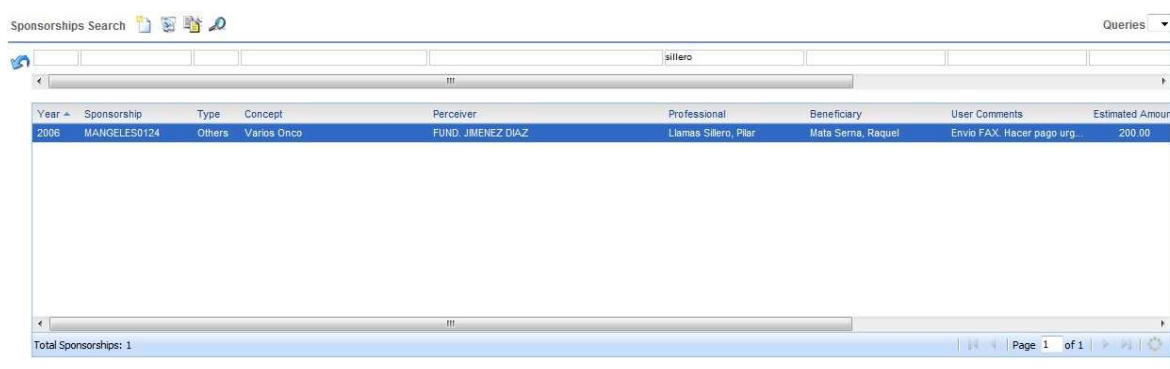


4.2.1.2 Buscador de colaboraciones

El requisito 2.3.1 cita así

“

La aplicación deberá contener una herramienta consistente en un grid paginado en el que se mostrarán las colaboraciones. Se incorporará una herramienta de búsqueda que permita discriminar por cualquier campo mostrado en el grid. Se mostrarán un máximo de diez registros en cada una de las páginas.



Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments	Estimated Amount
2006	MANGELES0124	Others	Varios Onco	FUND. JIMENEZ DIAZ	Llamas Sillero, Pilar	Mata Serna, Raquel	Envio FAX. Hacer pago urg.	200.00

La imagen anterior contiene el selector personal de queries, esta característica del interfaz se corresponde con el requerimiento 2.3.2



“

La aplicación debe permitir que los usuarios puedan crear consultas y almacenarlas. La aplicación mostrará al usuario conectado sus consultas almacenadas, al seleccionarla será ejecutada.

4.2.1.3 Quick launch

La gestión de la queries, tal y como se pedia en el requisito 2.3.4, se debía realizar a través de una herramienta llamada Quick Launch.

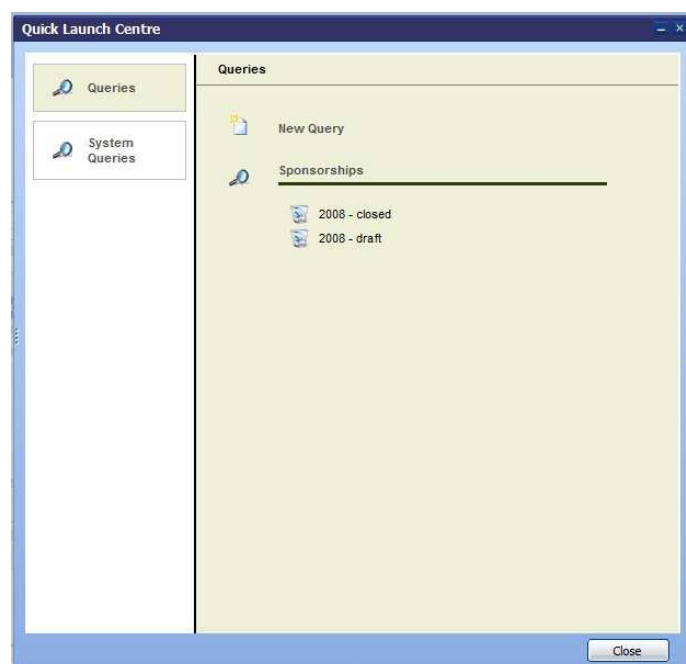
“

Se ofrecerá una herramienta de acceso rápido al mantenimiento y ejecución de *User Queries* y *System Queries*

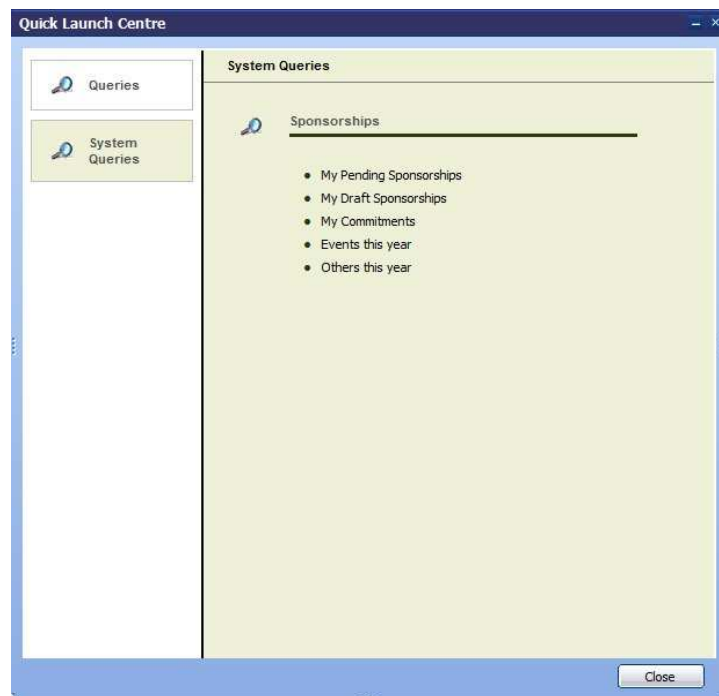
Se decide insertar un icono de acceso rápido debajo del menú de navegación. La idea del cliente es ir añadiendo nuevas opciones dentro de esta herramienta.



Al pulsar sobre el icono nos aparece el interfaz de gestión de user queries.

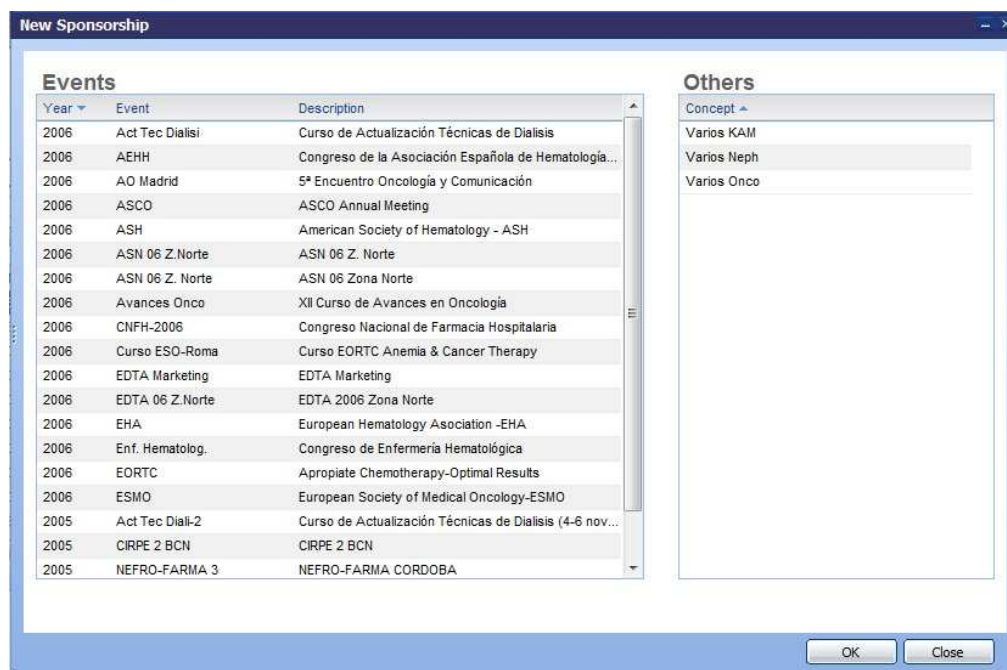


Desde el mismo interfaz tenemos acceso a System Queries



4.2.1.4 Crear colaboración

Antes de crear una colaboración debemos seleccionar el tipo. Estos tipos vienen predefinidos por el cliente, el usuario podrá seleccionar el que quiera de la lista.



El requerimiento 2.3.9 dice así.

La aplicación deberá contener las siguientes herramientas de mantenimiento de las colaboraciones: crear, borrar, copiar.

General Details			
Sponsorship ID:	<input type="text"/>	Products :	<input type="checkbox"/> Aranesp Onco <input type="checkbox"/> Aranesp Nefro <input type="checkbox"/> Neulasta <input type="checkbox"/> Mimpara <input type="checkbox"/> Neupogen
Concept:	Varios KAM		
Dept. Code: *	Spain Marketing Oncology ▾		
G/L Account SAP: *	Sales & Marketing Expense KOL & Brand Advocate (\$) ▾		
AC: *	a client ▾	Typology: *	
Approver: *	Alonso, Josecho ▾		
Status:	Draft		

Sponsorship Details:

Request Date: *	29/12/2008 (dd/mm/yyyy)	Total Estimated Amount:	0.00 €
Perceiver: *			
Beneficiary: *	<input type="text"/>	NIF:	<input type="text"/>
BO: *			
<input type="checkbox"/> Member		<input type="checkbox"/> Travel	
<input checked="" type="checkbox"/> Inscription	<div style="background-color: #cccccc; height: 40px;"></div>	<div style="background-color: #cccccc; height: 40px;"></div>	Departure Date: <input type="text"/>
PO:	Estimated Amount: 0.00 €	PO:	Estimated Amount: 0.00 €
<input type="checkbox"/> Miscellaneous	<div style="background-color: #cccccc; height: 40px;"></div>	<input type="checkbox"/> Accomodation	<div style="background-color: #cccccc; height: 40px;"></div>
PO:	Estimated Amount: 0.00 €	PO:	Estimated Amount: 0.00 €
<input type="checkbox"/> Other external services	<div style="background-color: #cccccc; height: 40px;"></div>	<input type="checkbox"/> Goods	<div style="background-color: #cccccc; height: 40px;"></div>
PO:	Estimated Amount: 0.00 €	PO:	Estimated Amount: 0.00 €
<input type="checkbox"/> Other	<div style="background-color: #cccccc; height: 40px;"></div>		Arrival Date: <input type="text"/> Departure Date: <input type="text"/>
PO:	Estimated Amount: 0.00 €		

User Comments:

El interfaz queda dividido en dos secciones, en la superior se informan las características más generales de la colaboración. En la inferior se informan detalles como el beneficiario y los conceptos.

4.2.1.6 Gestión de definiciones de evento

El requerimiento 2.3.10 decía los siguiente

“

La aplicación tendrá una herramienta capaz de gestionar las definiciones de eventos.

El acceso a esta interfaz se encuentra dentro del apartado de administración. Desde este mismo interfaz podremos buscar, crear, editar y borrar las definiciones de evento.

Events Definition

Year	ID	Event	Description	Dept. Code	GL Account SAP	AC	Active
2006	DEV060008	Avances Onco	XII Curso de Avances en Oncología	10665	Sales & Marketing Expense KOL & Bra...	aidak	Yes
2006	DEV060009	CRPE BCN06	CRPE BARCELONA 9-10 JUNIO 2006	10666	Symposia / Workshops Expense	carlas	No
2006	DEV060010	CRPE MAD06	CRPE MADRID (2-3 JUNIO 2006)	10666	Symposia / Workshops Expense	carlas	No
2006	DEV060011	CNFH-2006	Congreso Nacional de Farmacia Hospitalaria	10665	Sales & Marketing Expense KOL & Bra...	aidak	Yes
2006	DEV060012	Curso EORTC-Roma	Curso EORTC Anemia & Cancer Therapy	10665	Symposia / Workshops Expense	aidak	Yes
2006	DEV060013	CURSO PK	I Curso Farmacocinética para Nefrólogos	10666	Symposia / Workshops Expense	carlas	No
2006	DEV060014	EBMT - Hamburgo	32nd Annual Meeting of the European Group for Blood ...	10665	Symposia / Workshops Expense	cmirand	No
2006	DEV060015	EDTA Marketing	EDTA Marketing	10666	Symposia / Workshops Expense	angelesn	Yes
2006	DEV060016	EDTA 06 Z Norte	EDTA 2006 Zona Norte	10666	Sales & Marketing Expense KOL & Bra...	angelesn	Yes
2006	DEV060017	EHA	European Hematology Association -EHA	10665	Symposia / Workshops Expense	aidak	Yes

Total Definitions: 123

Event Definition Detail

Id: DEV060010 Event: CRPE MAD06 Year: 2006 Description: CRPE MADRID (2-3 JUNIO 2006)

Dept. Code: 10666 - Spain Marketing Nephrology % RC: 0.00

GL Account SAP: Symposia / Workshops Expense AC: Arias Gutiérrez-Acaltuno, Cristina

Typology: Active

Direct Expenses

Guide:

BO: To be defined in Sponsorships

Perceiver Type: Account

	Alias	Mail	PO	Estimated Amount
<input checked="" type="checkbox"/> Inscription	Cristina Arias	david.maiz@gmail.com,emunoz	6001338	
<input checked="" type="checkbox"/> Travel	Beneandco, SL	david.maiz@gmail.com,emunoz	6001654	
<input checked="" type="checkbox"/> Miscellaneous	Beneandco, SL	david.maiz@gmail.com,emunoz	6001654	
<input checked="" type="checkbox"/> Accomodation	Beneandco, SL	david.maiz@gmail.com,emunoz	6001654	
<input type="checkbox"/> Other external services				
<input type="checkbox"/> Goods				
<input type="checkbox"/> Other				

Indirect Expenses

BO: Migrated 100.00 %

Perceiver: Migrated

	Alias	Mail	PO	Estimated Amount
<input type="checkbox"/> Goods				
<input type="checkbox"/> External services				
<input type="checkbox"/> Infrastructure				
<input checked="" type="checkbox"/> Other				

El interfaz consta de dos partes, desde la primera podremos seleccionar la definición que queremos modificar. Al hacerlo aparecerán los datos de la misma en la parte inferior, donde la podremos editar.

4.2.1.7 Gestión de Tipologías

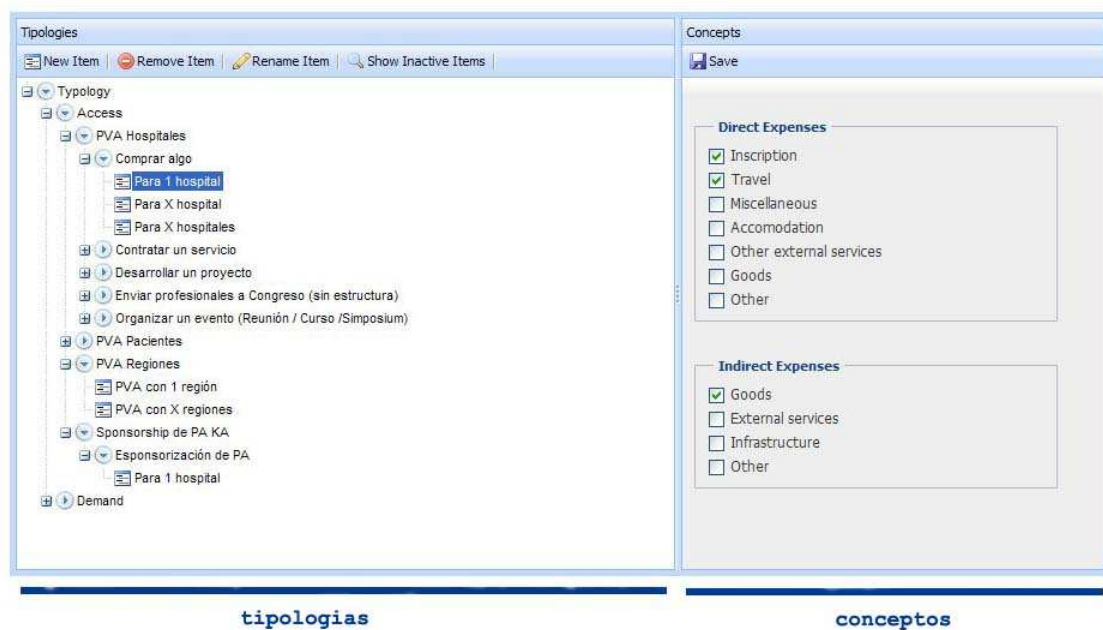
El requerimiento 2.3.11 decía los siguiente

“

Se requiere la existencia de una herramienta que gestione las tipologías de las colaboraciones.

Siguiendo la línea de la aplicación, se construyó un único interfaz que incorporase todas las operaciones de la gestión de tipologías.

El interfaz se divide en dos parte, en la izquierda se navega a través de las tipologías. La derecha muestra los conceptos de la tipología seleccionada.

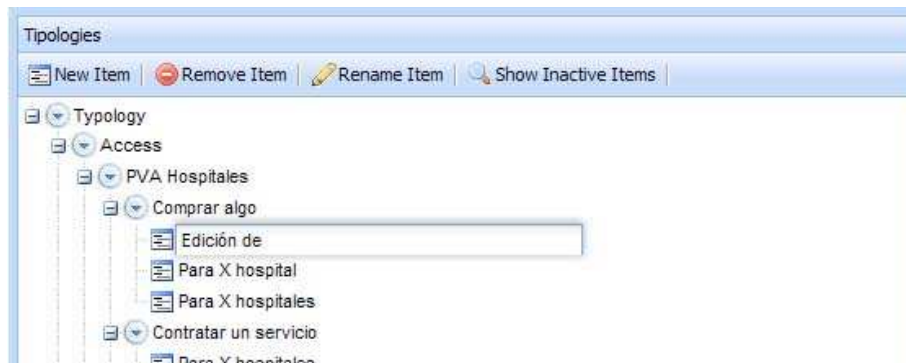


La navegación de las tipologías se realiza en forma de árbol, para hacerlo se aprovecharon las funcionalidades que ofrecía **Ext JS**.

El interfaz ofrece una barra de herramientas dinámica que activa o desactiva las operaciones disponibles dependiendo del estado (activo, inactivo) o el tipo (rama, hoja) de la tipología que seleccionemos.



Al asignar un nombre a una nueva tipología, ya sea porque es nueva o porque se quiere renombrar, se utiliza un sistema que permite editarlo directamente sobre el árbol.



El interfaz distingue una tipología activa de otra inactiva mostrando la inactiva en un tono más grisáceo.



4.3 Pruebas

Las Pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software.

Las pruebas de software se integran dentro de las diferentes fases del Ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Las pruebas de software, testing o beta testing es un proceso usado para identificar posibles fallos de implementación, calidad, o usabilidad de una aplicación.

Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Únicamente un proceso de verificación formal puede probar que no existen defectos.

Una definición de "testing" es: *proceso de evaluación de un producto desde un punto de vista crítico*, donde el "tester" (persona que realiza el testeo) somete el producto a una serie de acciones inquisitivas, y el producto responde con su comportamiento como reacción. Por supuesto, nunca se debe testear el software en un entorno de producción. Es necesario testear los nuevos programas en un entorno de pruebas separado físicamente del de producción.

Distinguiremos entre errores de programación (o "bugs") y defectos de forma. En un defecto de forma, el programa no realiza lo que el usuario espera. Por el contrario, un error de programación puede describirse como un fallo en la semántica de un programa de ordenador. Éste podría presentarse, o no, como un defecto de forma si se llegan a dar ciertas condiciones de cálculo.

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costos. Hoy en día es crucial verificar y evaluar la calidad de lo construido de modo de minimizar el costo de su reparación. Mientras antes se detecte una falla, más barato es su corrección.

El proceso de prueba es un proceso técnico especializado de investigación que requiere de profesionales altamente capacitados en lenguajes de desarrollo, métodos y técnicas de testeo y herramientas especializadas. El conocimiento que debe manejar un ingeniero de prueba es muchas veces superior al del desarrollador de software.

El proceso de pruebas al que fue sometida la aplicación fue el siguiente:

- 1.- Pruebas unitarias.
- 2.- Pruebas de integración.
- 3.- Pruebas de validación.

Finalmente, tras haber superado las pruebas, se realizó un test de usabilidad.

4.3.1 Pruebas unitarias

La ejecución de las pruebas unitarias se efectúa probando los componentes por separado. La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto

Este tipo de pruebas se han encargado de probar todas las funcionalidades.

4.3.2 Pruebas de integración

Es la fase del testeo de software en la cual módulos individuales de software son combinados y testeados como un grupo. Son las pruebas posteriores a las pruebas unitarias y preceden el testeo de sistema.

Básicamente, consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

4.3.3 Pruebas de validación

Las **pruebas de validación** en la ingeniería de software son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido.

Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales. La pregunta a realizarse es: ¿Es esto lo que el cliente quiere?.

Para ello el cliente desarrollo un conjunto de pruebas que validaban las funcionalidades de la aplicación.

4.3.4 Test de usabilidad

Es una prueba de observación con usuarios reales mientras utilizan y realizan tareas con el producto. Es una manera directa de obtener información de los usuarios, analizando más lo que hacen que las opiniones.

La usabilidad se refiere a la facilidad o nivel de uso, es decir, al grado en el que el diseño de un objeto facilita o dificulta su manejo.

Características

Los principios básicos en los que se basa la usabilidad son:

1.- Facilidad de Aprendizaje, facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema. Está relacionada con la predicibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.

2.- Flexibilidad, relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.

3.- Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

Beneficios

Entre los principales beneficios de la usabilidad encontramos:

- Reducción de los costes de aprendizaje.
- Disminución de los costes de asistencia y ayuda al usuario.
- Optimización de los costes de diseño, rediseño y mantenimiento.
- Aumento de la tasa de conversión de visitantes a clientes de un sitio web.
- Mejora la imagen y el prestigio.
- Mejora la calidad de vida de los usuarios, ya que reduce su estrés, incrementa la satisfacción y la productividad.

4.3.4.1 *Proceso*

El proceso de realización del test de usabilidad siguió los siguientes pasos:

- 1.- Definir los objetivos e intereses a evaluar.
- 2.- Identificar perfiles de usuarios y reclutar participantes.
- 3.- Preparación y realización de test.
- 4.- Análisis de resultados y propuesta de soluciones.

Intereses a evaluar

Nos interesa evaluar las funcionalidades más importantes, que serán:

- Crear y solicitar una colaboración
- Buscar colaboración en grid
- Crear definición de evento
- Crear tipología
- Modificar conceptos de una tipología
- Borrar tipología
- Renombrar tipología

Perfiles de usuario

Nos interesa probar la aplicación con el máximo número de usuarios posible y de todos los perfiles. El cliente nos consiguió un grupo de empleados de diferentes roles para realizar el test.

Nombre	Rol
Toni	Administrador
Maria	Supervisora
Aida	Supervisora
Josep	Representante
Nuria	Representante

Preparación y realización de test

Para la realización de los test el cliente nos consiguió una sala en sus dependencias. La prueba la realizó cada usuario por separado, la idea era recrear un entorno sin distracciones.

Los usuarios disponían de un máximo de veinte minutos para probar las funcionalidades. Durante la prueba el usuario debía ir anotando su experiencia en un formulario, además durante la prueba se grabó todo lo que ocurría en la pantalla, esto nos ayudó a ver que partes eran más costosas para los usuarios. Tras realizar el test se mantiene una charla con el usuario con el que se acaban de matizar los aspectos ha mejorar del producto.

El formulario entregado a cada usuario contenía las instrucciones de lo que debía hacer. Tras realizar cada funcionalidad debía rellenar un pequeño cuestionario que puntuaba con una nota del uno al cinco características como la estética, la velocidad y la facilidad de uso.

Todos los usuarios reclutados para el test realizaron todas las pruebas, a pesar de que muchos de ellos, debido a su rol, no tendrían acceso a algunas de las funcionalidades que estaban probando.

Crear y solicitar una colaboración

1.- Crear y solicitar una colaboración tipo otros con los siguientes datos

Productos: Producto1, Producto3.

Tipología: Acces/PVA Hospitales/comprar algo/Para x Hospitales

Perceptor: Cuenta/Hosp. Vall d'Hebron

Profesional: Sergio García

Beneficiario: Sergio García

BO: SPA - BU KAM SF Field Position Norte-05

POs: 1111111111

¿Has logrado completar la tarea?	SI	NO
¿Te ha resultado sencillo?	1 2 3 4 5	
¿Te gusta la estética?	1 2 3 4 5	

Buscar colaboración en el Grid

2.- Buscar las siguientes colaboraciones a partir de estos datos

a) Hospital: Miguel Server, Profesional: Juan Perez, Año: 2008

b) Identificador: TGARCIA0001

c) Estado: cerrada, Profesional: José Manuel Martínez, Hospital: La paz

¿Has logrado completar las tareas?	SI	NO
¿La búsqueda es lo bastante rápida?	1 2 3 4 5	
¿Te resulta útil la búsqueda a través del grid?	1 2 3 4 5	

Crear definición de evento

3.- Crea una definición de evento con los siguientes datos

Evento: Test

Año: 2008

Descripción: Test desc.

Productos: Producto1, Producto3.

Tipología: Acces/PVA Hospitales/comprar algo/Para x Hospitales

POs: 1111111111

Costes estimados: 100 €

Proveedor: proveedorTest

Costes directos

BO: SPA - BU KAM SF Field Position Norte-01

Tipo perceptor: Cuenta

Costes indirectos

BO: SPA - BU KAM SF Field Position Norte-02

Perceptor: Región/Cataluña/100%

¿Has logrado completar la tarea?	SI	NO
¿Te ha resultado sencillo?	1 2 3 4 5	
¿Te gusta la estética?	1 2 3 4 5	

Gestionar tipologías

4.- Realiza las siguientes acciones sobre el gestor de tipologías

- Crea una tipología de nombre test en: Acces/PVA Pacientes/Contratar un servicio
- Marca el concepto directo Travel, el indirecto Goods y guardalos.
- Renombra la tipología *test* por *test renombrado*
- Elimina la tipología *test renombrado*

¿Has logrado completar las tareas?	SI	NO
¿Te ha resultado sencillo?	1 2 3 4 5	
¿Te gusta la estética?	1 2 3 4 5	

4.3.4.2 Resultados y conclusiones

Tras haber recopilado toda la información: Cuestionarios, videos y charlas. Llega el momento de analizarla.

Videos

Tras analizar los vídeos descubrimos dos fallos graves. Al solicitar la colaboración del test 1, habiendo añadido unos costes superiores al millar, se producía un error que impedía al usuario terminar la prueba (*usuario Aída*)

El resto de videos analizados no mostraron nada fuera de lo común.

Charlas

Sin duda las charlas fue uno de los elemento que más información valiosa nos proporcionó. Tras analizarlas descubrimos que la mayoría de los usuarios coincidían en determinados aspectos estructurales de los formularios. Estos aspectos fueron tenidos en cuenta y corregidos para adecuarlos al gusto de la mayoría de los usuarios tratados.

Cuestionarios

El análisis de los cuestionarios arrojó datos muy favorables sobre el trabajo realizado. La mayoría opinaba bien o muy bien sobre las características encuestadas. Estos datos fueron presentados al cliente, el cual se dio igualmente por satisfecho con el resultado.

Usuario	Prueba	Pregunta	Respuesta
TONI - Administrador	P1	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	4
		¿Te gusta la estética?	3
	P2	¿Has logrado completar la tarea?	SI
		¿La búsqueda es lo bastante rápida?	4
		¿Te resulta útil la búsqueda a través del grid?	5
	P3	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	3
	P4	¿Has logrado completar la tarea?	SI
		¿Te ha resultado sencillo?	5
		¿Te gusta la estética?	5

Usuario	Prueba	Pregunta	Respuesta
Maria - Supervisora	P1	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	3
	P2	¿Has logrado completar la tarea?	SI
		¿La búsqueda es lo bastante rápida?	5
		¿Te resulta útil la búsqueda a través del grid?	5
	P3	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	4
	P4	¿Has logrado completar la tarea?	SI
		¿Te ha resultado sencillo?	4
		¿Te gusta la estética?	5

Usuario	Prueba	Pregunta	Respuesta
Aída - Supervisora	P1	¿Has logrado completar la tarea?	NO
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	4
	P2	¿Has logrado completar la tarea?	SI
		¿La búsqueda es lo bastante rápida?	5
		¿Te resulta útil la búsqueda a través del grid?	5
	P3	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	3
	P4	¿Has logrado completar la tarea?	SI
		¿Te ha resultado sencillo?	4
		¿Te gusta la estética?	5

Usuario	Prueba	Pregunta	Respuesta
Josep - Representante	P1	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	4
	P2	¿Has logrado completar la tarea?	SI
		¿La búsqueda es lo bastante rápida?	4
		¿Te resulta útil la búsqueda a través del grid?	5
	P3	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	3
	P4	¿Has logrado completar la tarea?	SI
		¿Te ha resultado sencillo?	4
		¿Te gusta la estética?	5

Usuario	Prueba	Pregunta	Respuesta
Nuria - Representante	P1	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	4
		¿Te gusta la estética?	4
	P2	¿Has logrado completar la tarea?	SI
		¿La búsqueda es lo bastante rápida?	4
		¿Te resulta útil la búsqueda a través del grid?	5
	P3	¿Has logrado completar la tarea?	SÍ
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	5
	P4	¿Has logrado completar la tarea?	SI
		¿Te ha resultado sencillo?	3
		¿Te gusta la estética?	5

5 • Estudio temporal y económico

En éste capítulo se va a realizar un estudio del temporal y el coste económico derivado del proyecto.

Para poder analizar el coste económico derivado de desarrollar el proyecto, es necesario que previamente estén cuantificados el número de horas y el tipo de recursos que se han necesitado para desarrollarlo. Así que, antes de poder realizar el estudio económico, será necesario realizar un estudio temporal del proyecto.

Contenidos de la sección

5.1 Estudio temporal inicial

5.2 Estudio temporal real

5.3 Desviación temporal

5.4 Estudio económico

5.1 Estudio temporal inicial

Después de haber logrado alcanzar los objetivos iniciales del proyecto y de satisfacer todos los requisitos del sistema que se habían detectado, es el momento de nombrar y describir las diferentes tareas que han sido necesarias para lograrlo.

Para desarrollar un estudio temporal de un proyecto, es necesario realizar una planificación temporal inicial del proyecto y una planificación temporal real del proyecto. De este modo el estudio temporal podrá determinar la desviación temporal derivada de su desarrollo.

La planificación temporal inicial de un proyecto, realiza una estimación del tiempo que será necesario para finalizar cada una de las fases del proyecto.

En cambio la planificación temporal real, cuantifica el tiempo que ha sido necesario para finalizar cada una de las fases del proyecto.

La desviación temporal de un proyecto es la diferencia entre la planificación temporal real del proyecto y la planificación temporal inicial del proyecto.

Hacer una buena planificación temporal inicial de las tareas que se van a realizar, asignando a cada tarea una duración y los recursos que serán necesarios para finalizarla, aumenta las posibilidades de realizar con éxito un proyecto.

El hecho de realizar una planificación temporal inicial aporta al desarrollo del proyecto las siguientes ventajas:

- **Optimiza la tarea de asumir compromisos** de tiempo y fechas de finalización de cada una de las fases del proyecto.
- **Mejora el seguimiento y la supervisión del proyecto**, haciendo más sencilla y fiable la tarea de identificar desviaciones y retardos en las fechas de entrega del proyecto.
- **Optimiza la toma de decisiones**, ya que el hecho de facilitar el seguimiento y la supervisión del proyecto, hace más sencilla y efectiva la labor de tomar decisiones.

A continuación se mostrará la planificación inicial de las tareas a desarrollar y el tiempo estimado necesario para finalizarlas.

Para realizar esta planificación temporal, se ha considerado que la jornada laboral es de 8 horas al día.

Nombre	Fecha de inicio	Fecha de fin	Duración
<input type="checkbox"/> Especificación	29/09/08	15/10/08	12
Definición especificaciones + doc. funcional	29/09/08	11/10/08	10
Modelo Conceptual	13/10/08	15/10/08	2
<input type="checkbox"/> Diseño	15/10/08	29/10/08	10
Análisis tecnológico	15/10/08	16/10/08	1
Diseño operaciones sistema	16/10/08	29/10/08	9
Construcción BD	29/10/08	1/11/08	3
<input type="checkbox"/> Mantenimiento tipologías	3/11/08	14/11/08	9
Implementación operaciones dominio	3/11/08	6/11/08	3
Implementación Interfaz	6/11/08	14/11/08	6
<input type="checkbox"/> Mantenimiento definición de eventos	14/11/08	26/11/08	8
Implementación operaciones dominio	14/11/08	21/11/08	5
Implementación interfaz	21/11/08	26/11/08	3
<input type="checkbox"/> Mantenimiento colaboraciones	26/11/08	26/12/08	22
<input type="checkbox"/> Inscritas a evento	26/11/08	11/12/08	11
Implementación operaciones dominio	26/11/08	4/12/08	6
Implementación interfaz	4/12/08	11/12/08	5
<input type="checkbox"/> No inscritas	11/12/08	26/12/08	11
Implementación operaciones dominio	11/12/08	19/12/08	6
Implementación interfaz	19/12/08	26/12/08	5
Pruebas	29/12/08	3/01/09	5



5.2 Estudio temporal real

A continuación se mostrará la planificación real de las tareas desarrolladas y el tiempo que ha sido necesario para finalizarlas.

Para realizar esta planificación temporal, se ha considerado que la jornada laboral es de 8 horas al día.

Nombre	Fecha de inicio	Fecha de fin	Duración
<input type="checkbox"/> Especificación	29/09/08	17/10/08	14
Definición especificaciones + doc. funcional	29/09/08	15/10/08	12
Modelo Conceptual	15/10/08	17/10/08	2
<input type="checkbox"/> Diseño	17/10/08	1/11/08	11
Análisis tecnológico	17/10/08	18/10/08	1
Diseño operaciones sistema	20/10/08	1/11/08	10
Nombre : Análisis tecnológico			
Construcción BD	3/11/08	5/11/08	2
<input type="checkbox"/> Mantenimiento tipologías	5/11/08	19/11/08	10
Implementación operaciones dominio	5/11/08	8/11/08	3
Implementación Interfaz	10/11/08	19/11/08	7
<input type="checkbox"/> Mantenimiento definición de eventos	19/11/08	29/11/08	8
Implementación operaciones dominio	19/11/08	26/11/08	5
Implementación interfaz	26/11/08	29/11/08	3
<input type="checkbox"/> Mantenimiento colaboraciones	1/12/08	1/01/09	23
<input type="checkbox"/> Inscritas a evento	1/12/08	16/12/08	11
Implementación operaciones dominio	1/12/08	10/12/08	7
Implementación interfaz	10/12/08	16/12/08	4
<input type="checkbox"/> No inscritas	16/12/08	1/01/09	12
Implementación operaciones dominio	16/12/08	26/12/08	8
Implementación interfaz	26/12/08	1/01/09	4
Pruebas	1/01/09	7/01/09	4

5.3 Desviación temporal

Comparando la información temporal real y inicial del proyecto podemos obtener las desviaciones de cada una de sus etapas.

El siguiente cuadro muestra las desviaciones de cada una de las etapas, el análisis de estas desviaciones puede ser útil para comprender y corregir futuras previsiones temporales.

Etapas	Inicial	Real	Balance
Especificación	12	14	+2
Diseño	10	11	+1
Construcción BD	3	2	-1
Mantenimiento tipologías	9	10	+1
Mantenimiento definiciones evento	8	8	0
Mantenimiento colaboraciones	22	23	+1
Pruebas	5	4	-1
	69	72	+3

5.4 Estudio económico

En el estudio económico, se pretende cuantificar los gastos derivados del desarrollo del proyecto.

Para poder desarrollar un proyecto se pueden necesitar programadores, analistas, comerciales, jefes de proyecto y otros tipos de recursos humanos. También se pueden necesitar maquinaria informática, software y otros tipos de recursos.

De este modo, para poder cuantificar los gastos derivados de este proyecto deberemos tener en cuenta recursos humanos y materiales.

5.4.1 Recursos materiales

El desarrollo de este proyecto debía realizarlo una persona en las dependencias del cliente. Se trataba de un entorno aislado, por lo que Grubit tendría que proporcionar los recursos materiales necesarios para el desarrollo.

Dividiremos estos recursos entre recursos hardware y software.

Recursos Hardware

HARDWARE NECESARIO	COSTE (en euros)
<i>PC + monitor + periféricos estándar.</i>	1200 €
Total:	1200 €

Recursos Software

HARDWARE NECESARIO	COSTE (en euros)
<i>Windows XP Professional, con SP2</i>	150 €
<i>Microsoft Office 2003</i>	180€
<i>Microsoft Office Visio 2003</i>	200€
<i>Microsoft Office Project 2003</i>	900€
<i>Microsoft Visual SourceSafe</i>	80 €
Total:	1510€

5.4.2 Recursos humanos

Ha continuación desglosaremos los gastos derivados a los distintos perfiles humanos necesarios para la realización del proyecto.

Se han considerado dos perfiles necesarios para el proyecto

- **Analista**
- **Programador**

Las funciones del analista son las siguientes

- **Preparación**
- **Análisis de requisitos**
- **Análisis de requisitos**
- **Especificación**
- **Diseño**
- **Documentación**
- **Presentación del proyecto**
- **Prueba**

Las tareas del programador son las siguientes

- **Implementación**
- **Prueba**

El siguiente cuadro muestra el desglose de horas entre los distintos perfiles necesarios para el proyecto.

PERFIL	HORAS	COSTE HORA	TOTAL
<i>Analista</i>	216	15.63€	3376,08
<i>Programador</i>	344	10.42€	3584,48
Total			6960,56

Para calcular el coste por hora, se ha supuesto que el año tiene 240 días laborables y que la jornada laboral es de 8 horas.

5.4.3 Gastos totales

Los gastos derivados de este proyecto son los gastos dedicados a recursos materiales más los gastos dedicados a recursos humanos.

Gasto total	
<i>Recursos materiales</i>	2710
<i>Recursos humanos</i>	6960,56
Total	9670,56

6 • Conclusiones

En este capítulo expondremos las conclusiones obtenidas durante el desarrollo de este proyecto. Estas conclusiones están relacionadas tanto en la aplicación construida, como en las experiencias a nivel personal.

Contenidos de la sección

6.1 La aplicación

6.2 Nivel personal

6.1 La aplicación

Actualmente la aplicación se encuentra funcionando correctamente. Tras el proceso de implantación se pudo comprobar que las especificaciones iniciales se habían cumplido.

Esta aplicación esta siendo utilizada por más de cien usuarios. Estos se encuentran cómodos con la herramienta y las sensaciones generales son buenas.

Se puso a disposición de todos los usuarios una herramienta de envío de incidencias. Los errores enviados se fueron tratando conforme llegaban. Aproximadamente el 90% de las incidencias recibidas fueron desechadas por no tratarse de auténticos errores de la aplicación, si no de mejoras o cambios en la estética. Esta información es muy valiosa, ya que con ella podría confeccionar en un futuro una nueva oferta de mantenimiento evolutivo. Tal y como comentábamos en el capítulo de pruebas, es muy importante la colaboración de los futuros usuarios a la hora de confeccionar una herramienta. Al fin y al cabo va a ser para ellos.

No existe ninguna aplicación similar en el mercado. Esto hace de nuestra aplicación un producto informático pionero en la gestión de las colaboraciones médicas.

Una vez el proyecto ha sido integrado con éxito en la empresa, el contacto con el cliente sigue siendo activo. Este contacto no se perderá ya que la aplicación creada es un producto en expansión. A esto hay que unir a que estamos gestionando un negocio muy cambiante, y esto acabará afectando a las especificaciones iniciales.

6.2 A nivel personal

Finalizar este proyecto significa cerrar un ciclo de aprendizaje que comenzó en septiembre el año 2000. Un ciclo que condiciona tu vida profesional y social futura.

Los conocimientos técnicos y sociales que ha aportado el desarrollo de este proyecto son muchos.

A nivel intelectual he obtenido conocimientos de técnicas punteras que se están utilizando en el desarrollo del software. La gran cantidad de nuevas tecnologías que aparecen hace que con cada nuevo proyecto almacenemos información útil que nos servirá para los siguientes.

Es importante resaltar la experiencia en las relaciones entre las personas que participan en el desarrollo del proyecto. Creo que gran parte del éxito de un proyecto depende de la calidad de las relaciones entre las personas que lo rodean.

La mejor conclusión que se puede sacar tras la finalización de este proyecto es que, sin duda, ahora estoy mejor preparado que al iniciarlo.

7 ● Bibliografía

En este capítulo se enumerarán todas las fuentes de información que han sido consultadas durante el desarrollo de este proyecto final de carrera, y también las fuentes de información relacionadas que pueden ser del interés del lector de esta memoria.

Se clasificará a las fuentes de información según su procedencia: Internet o libros de texto.

Contenidos de la sección

7.1 Internet

7.2 Libros de texto

7.1 Internet

<http://www.csi.map.es/csi/metrica3>

Sitio web donde encontrar información acerca de la metodología METRICA Versión 3. Es una **metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información.**

<http://es.wikipedia.org/>

Enciclopedia de libre actualización.

<http://hillside.net/patterns>

Sitio web, donde encontrar información acerca de los patrones de diseño.

<http://www.uml.org/>

UML Resource Page.

<http://java.sun.com/>

Java Resource Page.

<http://webdeveloper.com/>

Sitio web de referencia para programadores de aplicaciones web.

<http://www.w3c.es/>

Sitio web del Consorcio World Wide Web.

<http://www.w3schools.com/>

Sitio web donde encontrar tutoriales de tecnologías web.

7.2 Libros de texto

Binder .- ***Testing Object-Oriented Systems. Models, Patterns and Tools***, Addison-Wesley, 2000.

Costal, D.; Sancho, M.R.; Teniente, E. .- ***Enginyeria del Software: Especificació (Transparències del curs)***, Edicions UPC, 2000.

Gamma, Helm; Johnson, Vlissides: ***Design Patterns***, Addison-Wesley, 1994.

Gómez, C.; Mayol, E.; Olivé, A.; Teniente, E.: ***Enginyeria del Software: Disseny I (Transparències del curs - 2a edició)***, Edicions UPC, 2001.

Jacobson, Ivar; Booch, Grady; Rumbaugh, James: ***The Unified Modeling Language Reference Manual (2a edició)***, Addison-Wesley, 2004.

Jacobson, Ivar; Booch, Grady; Rumbaugh, James: ***The Unified Software Development Process***, Addison-Wesley, 1999.

Kotonya, G.; Sommerville, I.: ***Requirements Engineering, Processes and Techniques***, John Wiley & Sons, 2000.

Larman, Craig: ***Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development***, Prentice Hall, 2005.

Larman, Craig: ***UML y patrones***, Pearson, 2003.

8 ● Anexo-I: Manual de usuario

Usaremos este capítulo para añadir el manual de usuario de la aplicación. En el se detalla el funcionamiento de todas las funcionalidades. El objetivo de este manual es que un usuario si conocimientos previos de la aplicación sea capaz de aprender a usarla sin problemas. El manual aquí descrito es el mismo que utiliza el cliente durante el periodo de formación de los nuevos usuarios.

Contenidos de la sección

- 8.1 Introducción*
- 8.2 Colaboraciones*
- 8.3 Estado de las colaboraciones*
- 8.4 Proceso de las colaboraciones*
- 8.5 Grid de colaboraciones*
- 8.6 Uso de filtros*
- 8.7 Crear una colaboración*
- 8.8 Operaciones*
- 8.9 Definición de eventos*

8.1 Introducción

Esta parte de Creta permite gestionar las colaboraciones (viajes, hoteles, material...) que se realizan a diversos profesionales de los centros con los que trabaja la compañía.

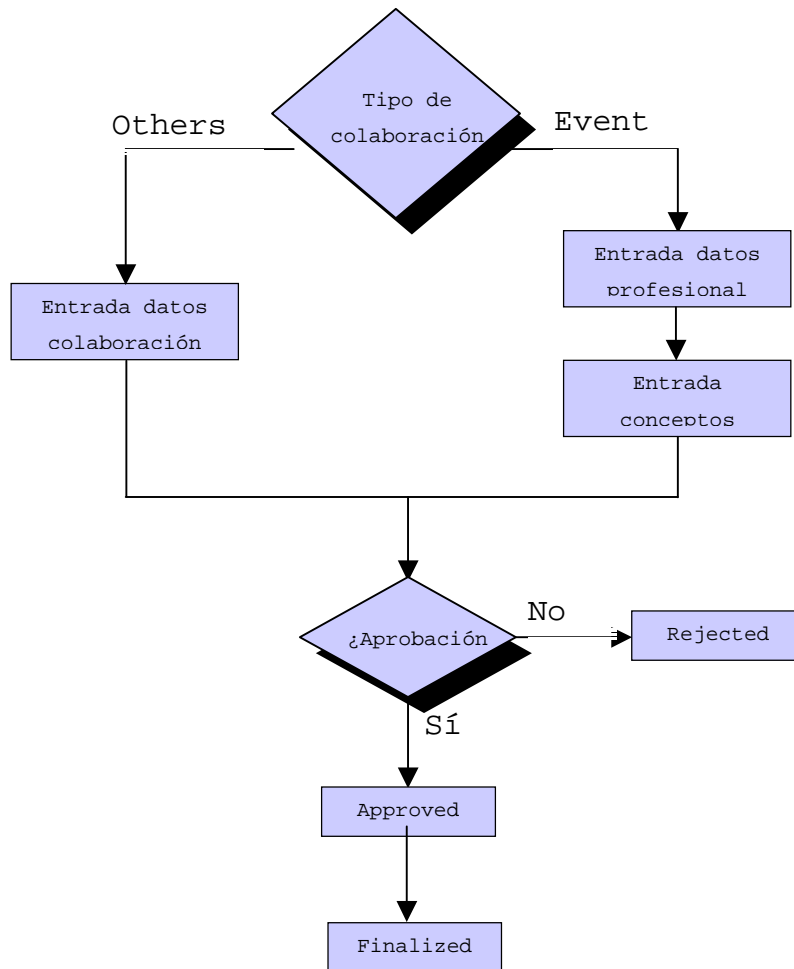
Se distinguen dos tipos de colaboraciones.

- Relacionadas con eventos (congresos)
- No relacionadas con eventos

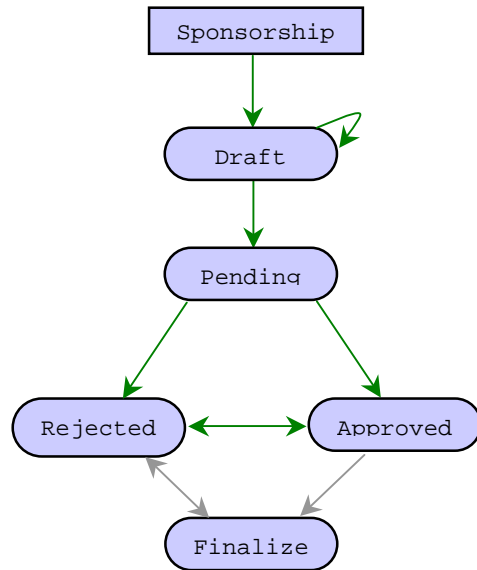
En el caso de los eventos se deben introducir los datos de los profesionales que se inscribirán en el evento, así como los detalles de inscripción, desplazamiento y varios que el profesional pueda necesitar.

En el resto de colaboraciones se debe introducir, además del beneficiario de la colaboración, sus detalles.

8.2 Colaboraciones



8.3 Estado de las colaboraciones



■ **Opciones válidas sólo para administradores**

Estado	Descripción
Draft	<p>Este estado permite introducir colaboraciones sin que entren en el proceso de aprobación.</p> <p>El usuario puede usar este estado para las colaboraciones que aún le faltan datos, para crear modelos de colaboraciones, etc...</p> <p>En este estado las colaboraciones se pueden modificar sin ninguna restricción.</p>
Pending	<p>Este estado indica las colaboraciones pendientes de aprobación.</p> <p>En este estado un Administrador puede modificar la colaboración.</p>
Approved	<p>Este estado designa las colaboraciones aprobadas.</p> <p>En este estado un Administrador puede modificar la colaboración.</p>
Rejected	<p>Este estado designa las colaboraciones que tras pasar el proceso de revisión se han denegado.</p> <p>En este estado el usuario que introdujo la colaboración la puede modificar para corregir la información incorrecta y solicitar de nuevo la aprobación.</p>
Finalized	<p>Este estado incluye todas las colaboraciones que en algún momento han sido aprobadas y ya han finalizado.</p> <p>En este estado un Administrador puede modificar la colaboración.</p>

8.4 Proceso de las colaboraciones

Al crear una colaboración se puede dejar en un estado Draft en caso de que sea necesario actualizar o modificar algún dato posteriormente.

Es posible modificar tantas veces cómo sea necesario cualquier colaboración en estado Draft.

Al pasar el estado a pendiente el supervisor realiza la revisión de la solicitud de colaboración pendiente y la aprueba o deniega.

En cualquiera de los dos casos, se envía un mail al usuario que dio de alta la colaboración.

En el caso que se deniegue la colaboración el usuario podrá hacer los cambios que crea oportuno y volver a solicitarla de forma que se pueda aprobar o denegar de nuevo.

8.5 Grid de colaboraciones

La pantalla aparece dividida en dos:

Sponsorships

Para cambiar de colaboración o evento se usan los cursores (\uparrow o \downarrow) para bajar o subir por la lista, de forma que automáticamente cambia la información en la parte inferior (Sponsorship Detail).

También se puede cambiar de colaboración con un clic directamente sobre la misma.

Sponsorship Detail

Muestra un detalle resumido de la colaboración o evento seleccionado en el grid de Sponsorships.






Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments	Estimated Amount
2008	AANDON0003	Event	Act Tec Dialisi	INST. CATALA REUMATOLOGIA	García García, Antonio Man...			0.00
2008	ACAMP0004	Event	ESMO	HOSP. DEL VALLE HEBRON General	Fort Ros, Joan	Montoro Ronsano, J.Bruno	x	1.00
2008	ALORCA0023	Others	Commitments Neupogen 2004	HOSP. VIRGEN DE LA LUZ (SESCAM)	González Joga, Beatriz	González Joga, Beatriz	User Comments:	10.00
2008	ALORCA0024	Others	Commitments Aranesa Onco 2006	HOSP. DEL VALLE HEBRON General				0.00
2008	ANIAA0051	Others	Libros y Revistas					0.00
2008	ANIAA0052	Others	Donaciones					0.00
2008	ANIAA0054	Others	Donaciones					0.00
2008	ANIAA0055	Others	Donaciones					0.00
2008	ANIAA0056	Others	Varios KAM			AGAFENE		0.00
2008	CARLOS0001	Event	PROVA	HOSP. DEL VALLE HEBRON - C.E.D. NEPHROS	Marco Cardona, Ferran	Marco Cardona, Ferran		57.00

Total Sponsorships: 210

Sponsorships Detail

Year: 2008	Perceiver: INST. CATALA REUMATOLOGIA	Estimated Amount: 0.00
Sponsorship: AANDON0003	Professional: García García, Antonio Manuel	Real Amount: 0.00
Scope: Account	Beneficiary:	REPIKAM: aandoni
Type: Event	User Comments:	DM:
Concept: Act Tec Dialisi	Mail:	AC: carlas
Status: Draft		

Los botones muestran las principales acciones a realizar en el grid.

Para ordenar la información se realiza un clic sobre el icono  de la columna que queremos ordenar. También es posible avanzar o retroceder a través de la lista de centros o acuerdos con sobre los iconos  y , ir a la primera o última página con  y , o introducir directamente el número de la página y pulsar la tecla "Enter".

SponsorshipsAdministrationLogoutv5.061013.1015

Quick Launch Centre

CarlosLast Visit: 29/12/2008

Sponsorships Search

Queries

Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments	Estimated Amount
2005	ALBSANCH0001	Others	Varios Neph	HOSP. CLI.U. LOZANO BLES	Martín Marín, Francisco	Martín Marín, Francisco	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0002	Others	Varios Neph	HOSP. CLI.U. LOZANO BLES	Martín Marín, Francisco	Martín Marín, Francisco	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0003	Others	Varios Neph	HOSP. MIGUEL SERVET	Arto Laborda, Maria Angeles	Arto Laborda, Maria Angeles	Hotel: Entrada 8.03.05 y Sa...	700.00
2005	ALBSANCH0004	Others	Varios Neph	HOSP. MIGUEL SERVET	Moreno Loshuertos, Silvia	Moreno Loshuertos, Silvia	Hotel: Entrada 8.03.05 y Sa...	700.00
2005	ALBSANCH0005	Others	Varios Neph	HOSP. MIGUEL SERVET	Gracia García, Olga	Gracia García, Olga	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0006	Others	Varios Neph	HOSPITAL MILITAR DE ZARAGOZA	Castilla Pérez, Jesús	Castilla Pérez, Jesús	Hotel: Entrada 06.03.05 y S...	1,126.01
2005	ALBSANCH0007	Event	EDTA	HOSP. CLI.U. LOZANO BLES	Ansón Manso, José Luis	Ansón Manso, José Luis	prueba prueba prueba prui...	3,200.01
2005	ALBSANCH0008	Event	EDTA Reps	HOSP. CLI.U. LOZANO BLES	Azuara Loscos, Manuel	Azuara Loscos, Manuel		3,200.01
2005	ALBSANCH0009	Others	Varios Neph	HOSP. MIGUEL SERVET	Gracia García, Olga	Gracia García, Olga	Hotel cercano a Pulgvert, e...	600.00
2005	ALBSANCH0010	Others	Varios Neph	HOSP. CLI.U. LOZANO BLES	Salanova Villanueva, Laura	Salanova Villanueva, Laura	Entrada el día 19 y salida el ...	900.00

Total Sponsorships: 4521Page 1 of 453

Sponsorships Detail

Year:
2005

Sponsorship:
ALBSANCH0001

Scope:
Account

Type:
Other

Concept:
Varios Neph

Status:
Rejected

Perceiver:
HOSP. CLI.U. LOZANO BLES

Professional:
Martín Marín, Francisco

Beneficiary:
Martín Marín, Francisco

User Comments:
Entrada 06.03.05 y salida 10.03.05

Mail:

Estimated Amount:
1,126.00

Real Amount:
0.00

REPIKAM:
albsanch


Dit:
rfores

AC:
adiment

8.6 Uso de filtros

La funcionalidad de los filtros permite definir condiciones sobre las colaboraciones y acotar la lista de resultados que se muestra en el listado.

Para poder aplicar filtros hay que introducir los criterios de búsqueda en los campos de filtro situados encima de las columnas correspondientes.

Para borrar los filtros se deberá pulsar el botón .

Sponsorships Search     Queries

Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments	Estimated Amount
2005	ALBSANCH0001	Others	Varios Neph	HOSP. CLIU. LOZANO BLESA	Martín Marín, Francisco	Martín Marín, Francisco	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0002	Others	Varios Neph	HOSP. CLIU. LOZANO BLESA	Martín Marín, Francisco	Martín Marín, Francisco	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0003	Others	Varios Neph	HOSP. MIGUEL SERVET	Arto Laborda, María Angeles	Arto Laborda, María Angeles	Hotel: Entrada 8.03.05 y Sal...	700.00
2005	ALBSANCH0004	Others	Varios Neph	HOSP. MIGUEL SERVET	Moreno Loshuertos, Sílvia	Moreno Loshuertos, Sílvia	Hotel: Entrada 8.03.05 y Sal...	700.00
2005	ALBSANCH0005	Others	Varios Neph	HOSP. MIGUEL SERVET	Gracia García, Olga	Gracia García, Olga	Entrada 06.03.05 y salida 1...	1,126.00
2005	ALBSANCH0006	Others	Varios Neph	HOSPITAL MILITAR DE ZARAGOZA	Castilla Pérez, Jesús	Castilla Pérez, Jesús	Hotel: Entrada 06.03.05 y S...	1,126.00
2005	ALBSANCH0007	Event	EDTA	HOSP. CLIU. LOZANO BLESA	Ansón Manso, José Luis	Ansón Manso, José Luis	prueba prueba prueba priu...	3,200.01
2005	ALBSANCH0008	Event	EDTA Reps	HOSP. CLIU. LOZANO BLESA	Azuara Loscos, Manuel	Azuara Loscos, Manuel		3,200.01
2005	ALBSANCH0009	Others	Varios Neph	HOSP. MIGUEL SERVET	Gracia García, Olga	Gracia García, Olga	Hotel cercano a Puigvert, e...	600.00
2005	ALBSANCH0010	Others	Varios Neph	HOSP. CLIU. LOZANO BLESA	Salanova Villanueva, Laura	Salanova Villanueva, Laura	Entrada el día 19 y salida el ...	900.00

Total Sponsorships: 4521 Page 1 of 453

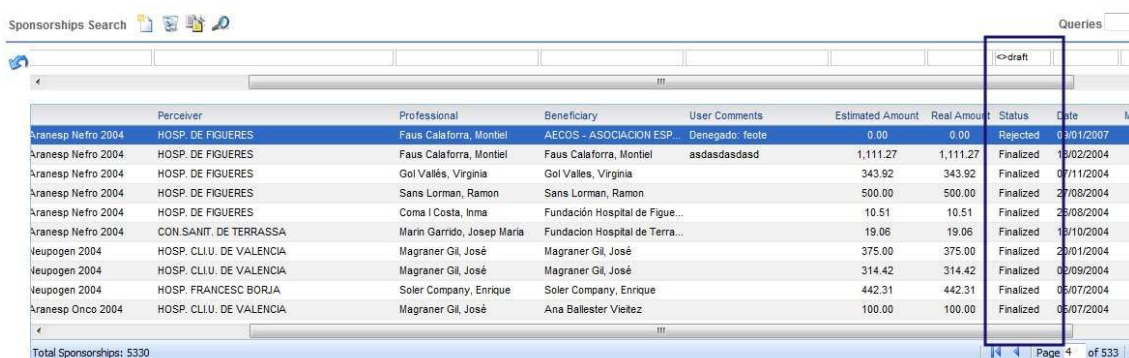
Operadores lógicos

* Carácter especial usado en la cadena de Hosp*Mar busca todas las colaboraciones cuyos centros búsqueda que substituye una subcadena de sean, por ejemplo, Hospital Gregorio Marañon y Hospital de caracteres. Sólo es necesario entre otros Zumarraga.	
? Carácter especial usado en la cadena de 200? busca todas las colaboraciones de los años 2000 a 2009 búsqueda que substituye un carácter.	
<> Valor distinto al indicado	<> Aprobado busca todas las colaboraciones no aprobadas
> Valor mayor que el indicado	> 2008 busca todas las colaboraciones que el año sea mayor que 2008
< Valor menor que el indicado	< 2008 busca todas las colaboraciones que el año sea menor que 2008
>= Valor mayor o igual que el indicado	>= 2008 busca todas las colaboraciones que el año sea mayor o igual que 2008
<= Valor menor o igual que el indicado	<= 2008 busca todas las colaboraciones que el año sea menor o igual que 2008
= Valor igual al indicado	= 2008 busca todas las colaboraciones del año 2004

Ejemplos

1.-Filtro simple: Mostrar todas las colaboraciones que no están en borrador.

Para obtener la lista de todas las colaboraciones que no sean del tipo borrador, es necesario escribir en el cuadro de edición encima de la columna Status el texto <>draft y pulsar la tecla *Enter*



Sponsorships Search

Queries

<>draft

	Perceiver	Professional	Beneficiary	User Comments	Estimated Amount	Real Amount	Status	Date	Ma
Aranesp Nefro 2004	HOSP. DE FIGUERES	Faus Calaforra, Montiel	AECOS - ASOCIACION ESP...	Denegado: feote	0.00	0.00	Rejected	01/01/2007	
Aranesp Nefro 2004	HOSP. DE FIGUERES	Faus Calaforra, Montiel	Faus Calaforra, Montiel	asdasdasdasd	1,111.27	1,111.27	Finalized	10/02/2004	
Aranesp Nefro 2004	HOSP. DE FIGUERES	Gol Vallés, Virginia	Gol Valles, Virginia		343.92	343.92	Finalized	07/11/2004	
Aranesp Nefro 2004	HOSP. DE FIGUERES	Sans Lorman, Ramon	Sans Lorman, Ramon		500.00	500.00	Finalized	27/08/2004	
Aranesp Nefro 2004	HOSP. DE FIGUERES	Coma I Costa, Inma	Fundación Hospital de Figue...		10.51	10.51	Finalized	20/08/2004	
Aranesp Nefro 2004	CON.SANIT. DE TERRASSA	Marin Garrido, Josep Maria	Fundacion Hospital de Terra...		19.06	19.06	Finalized	10/10/2004	
Veupogen 2004	HOSP. CLIU. DE VALENCIA	Magraner Gil, José	Magraner Gil, José		375.00	375.00	Finalized	20/01/2004	
Veupogen 2004	HOSP. CLIU. DE VALENCIA	Magraner Gil, José	Magraner Gil, José		314.42	314.42	Finalized	09/09/2004	
Veupogen 2004	HOSP. FRANCESC BORJA	Soler Company, Enrique	Soler Company, Enrique		442.31	442.31	Finalized	06/07/2004	
Aranesp Onco 2004	HOSP. CLIU. DE VALENCIA	Magraner Gil, José	Ana Ballester Viletez		100.00	100.00	Finalized	06/07/2004	

Total Sponsorships: 5330

Page 4 of 533

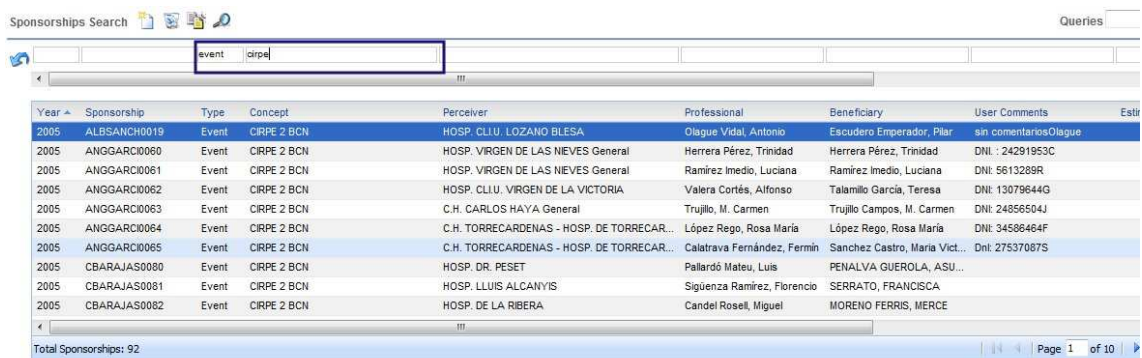
2.-Filtro combinado: Mostrar Todas las colaboraciones dónde el concepto sea CIRPE y sea del tipo evento

En este caso en la ventana de condiciones de filtro introduciremos los siguientes valores

Concept: CIRPE

Type: Event

El resultado final nos muestra todas las colaboraciones que cumplen las dos condiciones establecidas.



Sponsorships Search

event cirpe

Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments	Esti
2005	ALBSANCH0019	Event	CIRPE 2 BCN	HOSP. CLIU. LOZANO BLES	Olague Vidal, Antonio	Eacudero Emperador, Pilar	sin comentariosOlague	
2005	ANGGARC00060	Event	CIRPE 2 BCN	HOSP. VIRGEN DE LAS NEVES General	Herrera Pérez, Trinidad	Herrera Pérez, Trinidad	DNI: 24291953C	
2005	ANGGARC00061	Event	CIRPE 2 BCN	HOSP. VIRGEN DE LAS NEVES General	Ramírez Inedio, Luciana	Ramírez Inedio, Luciana	DNI: 5613289R	
2005	ANGGARC00062	Event	CIRPE 2 BCN	HOSP. CLIU. VIRGEN DE LA VICTORIA	Valera Cortés, Alfonso	Talamillo García, Teresa	DNI: 13079644G	
2005	ANGGARC00063	Event	CIRPE 2 BCN	C.H. CARLOS HAYA General	Trujillo, M. Carmen	Trujillo Campos, M. Carmen	DNI: 24856504J	
2005	ANGGARC00064	Event	CIRPE 2 BCN	C.H. TORRECARNENAS - HOSP. DE TORRECAR...	López Rego, Rosa María	López Rego, Rosa María	DNI: 34586464F	
2005	ANGGARC00065	Event	CIRPE 2 BCN	C.H. TORRECARNENAS - HOSP. DE TORRECAR...	Calatrava Fernández, Fermín	Sanchez Castro, Maria Vict...	Dni: 27537087S	
2005	CBARAJAS00080	Event	CIRPE 2 BCN	HOSP. DR. PESET	Pallardó Mateu, Luis	PENALVA GUEROLA, ASU...		
2005	CBARAJAS00081	Event	CIRPE 2 BCN	HOSP. LLUIS ALCANYIS	Sigüenza Ramírez, Florencio	SERRATO, FRANCISCA		
2005	CBARAJAS00082	Event	CIRPE 2 BCN	HOSP. DE LA RIBERA	Candel Rosell, Miguel	MORENO FERRIS, MERCE		

Total Sponsorships: 92

Page 1 of 10

3.-Uso de los operadores de substitución: Mostrar Todas las colaboraciones asociadas a la profesional "De Juan Arroyo, Ana".

Si deseamos buscar todas las colaboraciones asociadas a la profesional "De Juan Arroyo, Ana", no es necesario escribir el nombre entero. Únicamente indicando el primer apellido podemos obtener el resultado completo.

El asterisco intermedio indicaría, por ejemplo en "Juan*An" que queremos ver cualquier profesional que en la parte inicial del nombre aparezca el texto Juan (en este caso el primer apellido), permitiendo que la parte final contenga cualquier texto que empiece por An (en este caso Ana).

Sponsorships Search

Queries

Juan*An

Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comments
2004	RSANCLIM0178	Others	Commitments Aranesp Onco 2004	HOSPITAL DE GALDAKAO	De Juan Arroyo, Ana	De Juan Arroyo, Ana	
2004	RSANCLIM0181	Others	Commitments Aranesp Nefro 2004	HOSPITAL DE GALDAKAO	De Juan Arroyo, Ana	De Juan Arroyo, Ana	
2005	ANGGARC0046	Event	SEN	HOSPITAL VIRGEN DE LAS NIEVES General	Bravo Soto, Juan Antonio	Bravo Soto, Juan Antonio	NO PASAR NADA HA
2005	ANGGARC0070	Event	Masterclass BCN	HOSPITAL VIRGEN DE LAS NIEVES General	Bravo Soto, Juan Antonio	Bravo Soto, Juan Antonio	ver horarios avión pa
2005	ATOMIAS0030	Others	Libros y Revistas	CENTRO MEDICO SALUS BALEARES (Clinica Be...	Gasent Blesa, Juan Manuel	Gasent Blesa, Juan-Manuel	Juan Manuel Gasen B
2005	CDONADEU0010	Others	Servicios Profesionales	HOSPITAL GREGORIO MARAÑON	López Gómez, Juan Manuel	López Gómez, Juan Manuel	
2005	CDONADEU0054	Event	Masterclass BCN	HOSPITAL GREGORIO MARAÑON	López Gómez, Juan Manuel	López Gómez, Juan Manuel	
2005	CDONADEU0059	Others	Varios Neph	HOSPITAL GREGORIO MARAÑON	López Gómez, Juan Manuel	López Gómez, Juan Manuel	
2005	CGASQUE0018	Others	Varios Onco	HOSPITAL DE BARBASTRO	Linares Sanjuan, Francisco	Pavon Franco, Raquel	REALIZAR INSCRIPCIÓ
2005	FBLANCO0013	Others	Libros y Revistas	HOSPITAL GENERAL DE LA S.S. DE JEREZ	Torán Montserrat, Juan Daniel	Torán Montserrat, Juan Daniel	Por un año

Total Sponsorships: 41

Page 1 of 5



Las colaboraciones se identifican en diferentes colores según los siguientes criterios:

Negro: Approved, Finalized


Gris: Draft, Rejected

Verde: Pending

8.7 Crear una colaboración

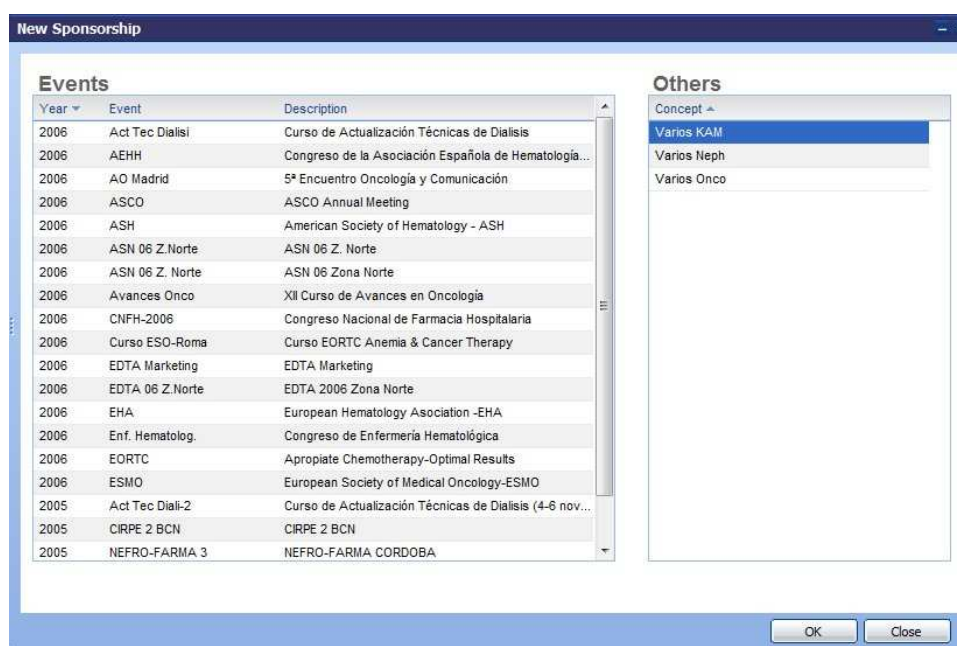
A continuación se muestra el procedimiento a seguir para crear una nueva colaboración.

8.7.1 Tipo Otros

Para crear una nueva colaboración se debe pulsar el botón **New Sponsorship**  que aparece en el menú sobre la lista de colaboraciones.



Para crear una **colaboración** de **Tipo Otros** simplemente se debe seleccionar el tipo asociado con la colaboración en el cuadro Others y pulsar el botón **Ok**.



General Details	
Sponsorship ID:	
Concept:	Vance KAM
Dept. Code:	Spain Marketing Oncology
G/L Account SAP:	Sales & Marketing Expense KOL & Brand Advocate (S)
AC:	a client
Approver:	Alonso, Josecho
Status:	Draft
Products:	<input type="checkbox"/> Aranesp Onco <input type="checkbox"/> Aranesp Refro <input type="checkbox"/> Neulasta <input type="checkbox"/> Mimpara <input type="checkbox"/> Neupogen <input type="checkbox"/> Vedibix
Typology:	

Sponsorship Details:

Request Date:	27/05/2008	(dd/mm/yyyy)	Total Estimated Amount:	0.00 €
Perceiver:				
Beneficiary:				
BO:				
<input type="checkbox"/> Member				
<input type="checkbox"/> Inscription				
PD:	Estimated Amount:			
		0.00 €		
<input type="checkbox"/> Miscellaneous				
PD:	Estimated Amount:			
		0.00 €		
<input type="checkbox"/> Other external services				
PD:	Estimated Amount:			
		0.00 €		
<input type="checkbox"/> Other				
PD:	Estimated Amount:			
		0.00 €		

MIF:		Departure Date:	
<input type="checkbox"/> Travel		Arrival Date:	
PD:	Estimated Amount:		
<input type="checkbox"/> Accomodation		Arrival Date:	
PD:	Estimated Amount:		
<input type="checkbox"/> Morables		Departure Date:	
PD:	Estimated Amount:		

La primera información a rellenar corresponde a los datos generales de la colaboración: AC, Cuenta, Supervisor, etc...

General Details



Sponsorship ID:

Concept:

Dept. Code: *

G/L Account SAP: *

AC: *

Approver: *

Status:

Products: *

☐ Aranesp Onco ☐ Aranesp Nefro

☐ Neulasta ☐ Mimpara

☐ Neupogen ☐ Vectibix

Typology: * 

A continuación se introduce la información de los productos asociados a la colaboración. De la selección que se realice dependerá la distribución en porcentajes para cada uno de ellos.

General Details



Sponsorship ID:

Concept:

Dept. Code: *

G/L Account SAP: *

AC: *

Approver: *

Status:

Products: *

☐ Aranesp Onco ☐ Aranesp Nefro

☐ Neulasta ☐ Mimpara

☐ Neupogen ☐ Vectibix

Typology: * 

Dentro del apartado de Sponsorship Details habrá que introducir la información que corresponda:

- Se introducirá la fecha de petición en el campo Request Date.
- Se seleccionará un perceptor desde la lista que nos aparece al pulsar el botón de la derecha del campo Perceiver.
- Se introducirá el beneficiario desde el campo Beneficiary.
- Por último, se introduce la información de detalle de la inscripción: Registration Details, Travel Details, Accommodation, ...

Sponsorship Details:

Request Date: 10/06/2008 (dd/mm/yyyy)

Perceiver:

Beneficiary:

BO:

☐ Member

☐ Inscription

PO: Estimated Amount: 0.00 €

☐ Miscellaneous

PO: Estimated Amount: 0.00 €

☐ Other external services

PO: Estimated Amount: 0.00 €

☐ Other

PO: Estimated Amount: 0.00 €

User Comments:

Total Estimated Amount: 0.00 €

NIF:

☐ Travel

PO: Estimated Amount: 0.00 €

☐ Accommodation

PO: Estimated Amount: 0.00 €

☐ Movables

PO: Estimated Amount: 0.00 €


Departure Date:

Arrival Date:

Arrival Date:

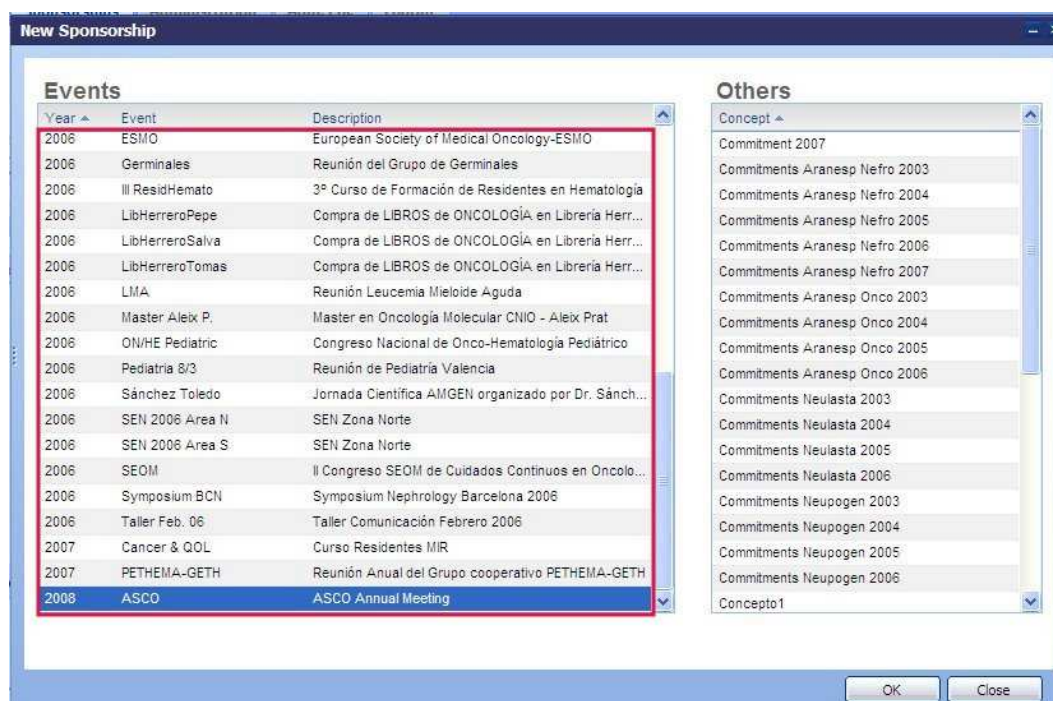
Departure Date:

8.7.2 Tipo Evento

Para crear una nueva colaboración se debe pulsar el botón **New Sponsorship**  que aparece en el menú sobre la lista de colaboraciones.



Para crear una colaboración de Tipo Evento simplemente se debe seleccionar el evento sobre el que queremos crear la colaboración sobre el listado Events y pulsar el botón **Ok**.



Dentro del apartado de Sponsorship Details habrá que introducir la información que corresponda:

- Se introducirá la fecha de petición en el campo Request Date.
- Se seleccionará un perceptor desde la lista que nos aparece al pulsar el botón de la derecha del campo Perceiver.
- Se introducirá el beneficiario desde el campo Beneficiary.

Por último, se introduce la información de detalle de la inscripción: Registration Details, Travel Details, Acommodation, ...

Sponsorship Details

Request Date: * (dd/mm/yyyy)

Perceiver: *

Professional: *

Beneficiary: *

BO: *

NIF:

NIF:

Total Estimated Amount: 0.00 €

Total Real Amount: 0.00 €

☐ Member

☒ Inscription

☒ Miscellaneous

☐ Other external services

☐ Other

PO: Estimated Amount: Real Amount:

PO: Estimated Amount: Real Amount:

PO: Estimated Amount: Real Amount:

PO: Estimated Amount: Real Amount:

User Comments:

☒ Travel

☒ Accomodation

☐ Movables

PO: Estimated Amount: Real Amount:

PO: Estimated Amount: Real Amount:

PO: Estimated Amount: Real Amount:

Supervisor Comments:

Departure Date:

Arrival Date:

Arrival Date:

Departure Date:

General Details




Event Def ID:
 G/L Account SAP:
☒ Aranesp Onco
 ☐ Aranesp Nefro
 Typology: ^{*} Demand

Concept:
 AC:
☒ Neulasta
 ☐ Mimpara
 Eventos

Dept. Code:
 Approver:
☐ Neupogen
 ☐ Vectibix
 Enviar profesionales a

Congreso sin Estructura
 Arngen
 De cualquier ámbito

General Details

Event Def ID: G/L Account SAP: ☒ Aranesq Grco ☐ Aranesq huetu

Concept: AC: ☒ Nuevaia ☐ Impara

Dept. Code: Approver: ☐ Neupreget ☐ Vectbox

Typology: ☐ Demand
☐ Evento
☐ Enviar profesionales a
☐ Congreso sin Estructura
☐ Aranes
☐ De cualquier otro

Participants

Sponsorship	Status	Perceiver	Beneficiary	Int	Total amount	Date	Details
Misculo.p001	Draft				0.0	27/06/2008	H

1 Draft 0 Pending 0 Approved 0 Finalized 0 Rejected Total Participants = 1

Sponsorship Details

Request Date:

Perceiver:

Beneficiary:

DO:

☐ Member ☒ Travel ☐

Total Estimated Amount: Total Real Amount:

Departure Date:

164

8.8 Operaciones

Sobre las colaboraciones podemos realizar diversas acciones, a continuación las detallamos

8.8.1 Operaciones sobre colaboraciones inscritas

Al crear o editar una colaboración inscrita a un evento nos aparecen las siguientes operaciones.

General Details


Event Def ID: DEV060006 G/L Account SAP: Sales & Marketing Expense KOL & Brand Advo
Concept: ASN 06 Z Norte AC: Navarro, Angels
Dept. Code: 10666 Approver: Alonso, Josecho


☐ Aranesp Onco ☒ Aranesp
☐ Neulasta ☐ Mimpara
☐ Neupogen

Participants



     

Sponsorship	Status	Perceiver	Beneficiary	Nif	Total
CNARCISO0235	Draft	HOSP. DEL VALLE HEBRON Trauma	Marsal Barri, Sara		0 €

Delete (Event) : Borra el Evento con todas sus inscripciones asociadas, siempre que todas ellas estén en estado Draft.

New : Crea una inscripción nueva.

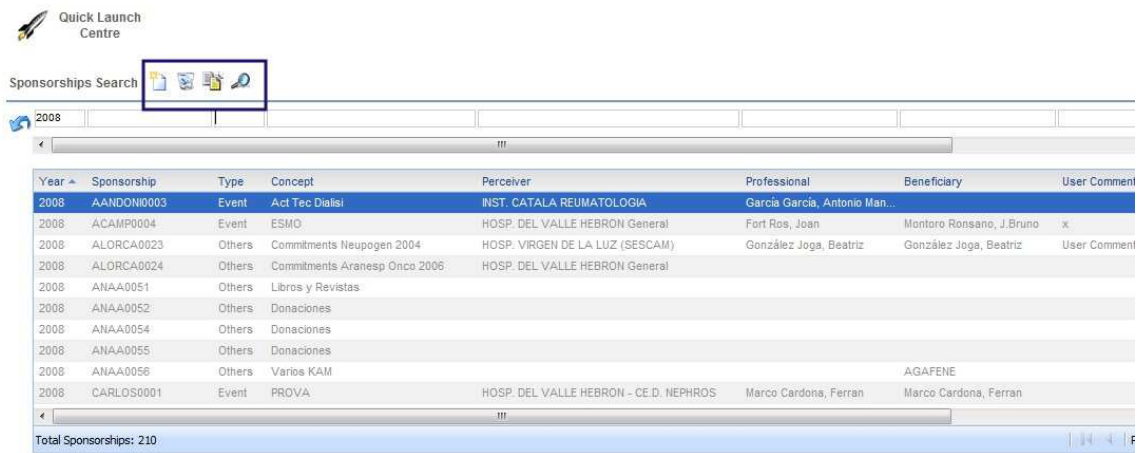
Delete : Elimina la inscripción, sólo se puede aplicar a inscripciones en estado Draft.

Submit  / **Submit all** : Guarda la inscripción seleccionada o todas (Submit All) y las deja en estado de pendiente de aprobación.

Print : Imprime la inscripción marcada en el grid.

8.8.2 Operaciones sobre el Grid

A continuación se describen las posibles operaciones sobre la lista de colaboraciones.





Year	Sponsorship	Type	Concept	Perceiver	Professional	Beneficiary	User Comment
2008	AANDON0003	Event	Act Tec Dialisi	INST. CATALA REUMATOLOGIA	García García, Antonio Man...		
2008	ACAMP0004	Event	ESMO	HOSP. DEL VALLE HEBRON General	Fort Ros, Joan	Montoro Ronsano, J. Bruno	x
2008	ALORCA0023	Others	Commitments Neupogen 2004	HOSP. VIRGEN DE LA LUZ (SESCAM)	González Joga, Beatriz	González Joga, Beatriz	User Comment
2008	ALORCA0024	Others	Commitments Aranesp Onco 2006	HOSP. DEL VALLE HEBRON General			
2008	ANAA0051	Others	Libros y Revistas				
2008	ANAA0052	Others	Donaciones				
2008	ANAA0054	Others	Donaciones				
2008	ANAA0055	Others	Donaciones				
2008	ANAA0056	Others	Varios KAM			AGAFENE	
2008	CARLOS0001	Event	PROVA	HOSP. DEL VALLE HEBRON - CE.D. NEPHROS	Marco Cardona, Ferran	Marco Cardona, Ferran	


Total Sponsorships: 210


Además, al pulsar sobre el grid con el botón derecho del ratón se nos mostrará el siguiente menú contextual.

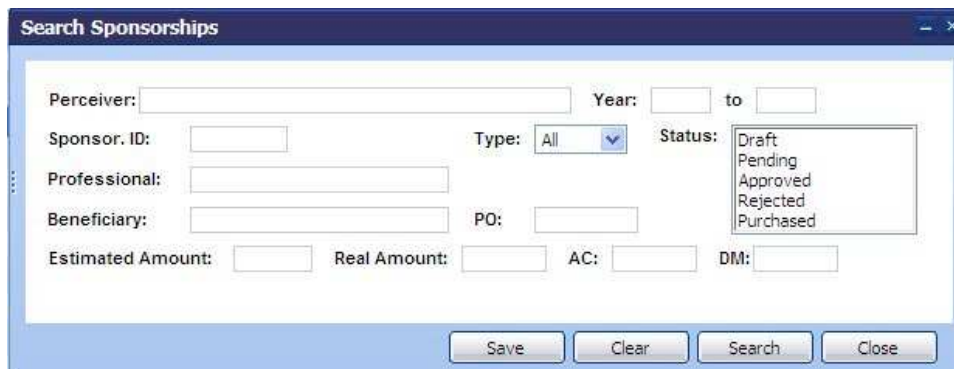


Para crear una nueva colaboración se realiza un clic sobre el botón  o la opción **New Sponsorship** del menú.

Para borrar una colaboración se hace un clic sobre la colaboración que deseamos eliminar y pulsamos el botón  o en el menú la opción **Delete Sponsorship**.

Para copiar una colaboración se hace un clic sobre la colaboración deseada y pulsamos sobre el botón  o en el menú escogemos la opción **Copy Sponsorship**. Esto nos creará una copia exacta de la colaboración o inscripción, y nos enviará al formulario de edición para que podamos modificarla.

Se puede también realizar una búsqueda con varios parámetros, haciendo un clic en el botón **Search** .



The image shows a 'Search Sponsorships' dialog box with the following fields and controls:

- Perceiver:
- Year: to
- Sponsor. ID:
- Type: (dropdown arrow)
- Status:
 - ☐ Draft
 - ☐ Pending
 - ☐ Approved
 - ☐ Rejected
 - ☐ Purchased
- Professional:
- Beneficiary:
- PO:
- Estimated Amount:
- Real Amount:
- AC:
- DM:

Buttons at the bottom:

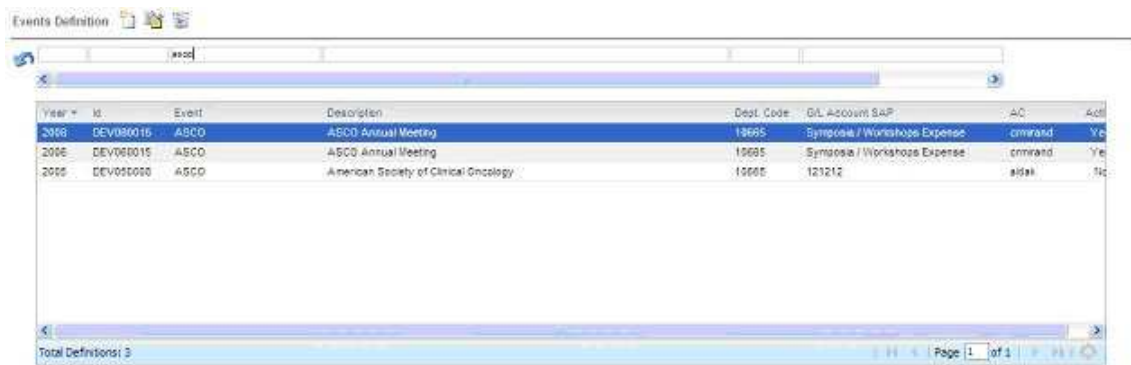
8.9 Definición de eventos

A continuación se explicará la creación de una nueva definición de evento.

Cuando se crea una nueva definición de evento se están introduciendo los datos que heredarán los eventos con dicha definición.

El interfaz de definición de eventos está dividido en dos partes: "Event definition" y "Event definition detail".

En la primera disponemos de un grid donde aparecerán todas las definiciones de evento disponibles.



The screenshot shows a window titled "Events Definition" with a search bar containing "asco". Below the search bar is a table with the following data:

Year	ID	Event	Description	Dept. Code	GL Account SAP	AC	Act
2008	DEV080015	ASCO	ASCO Annual Meeting	10695	Symposia / Workshops Expense	command	Yes
2008	DEV080015	ASCO	ASCO Annual Meeting	10695	Symposia / Workshops Expense	command	Yes
2008	DEV080015	ASCO	American Society of Clinical Oncology	10695	121212	command	Yes

At the bottom of the window, it says "Total Definitions: 3" and "Page 1 of 1".

Las operaciones que podremos realizar sobre este grid son las siguientes.

New : Crea una nueva definición de evento

Copy : Copia los datos de un evento a una nueva

Delete : Borra una definición.

En la segunda parte, Event definition detail, nos aparecerán los detalles del evento que estamos editando.

Event Definition Detail

Id: DEV080015 Event: ASCO Year: 2008 Description: ASCO Annual Meeting

Dept. Code: 10665 - Spain Marketing Oncology % RC: 0.00

GIL Account SAP: Symposia / Workshops Expense AC: Miranda, Cristina

Typology: Demand
Eventos
Enviar profesionales a Congreso sin Estructura Amgen
De cualquier ámbito ☒ Active

☐ Aranesp Nefro
☒ Aranesp Onco
☐ Mimpara
☒ Neulasta
☐ Neupogen
☐ Vectibix

Direct Expenses

Guide:

BO: ☐ To be defined in Sponsorships
SPA - BU KAIM Position 001 100.00 %

Perceiver Type: Region

	Alias	Mail	PO	Estimated Amount
<input checked="" type="checkbox"/> Inscription			0123456789	56.00
<input checked="" type="checkbox"/> Travel			0123456789	43.00
<input checked="" type="checkbox"/> Miscellaneous			0123456789	152,465.00
<input checked="" type="checkbox"/> Accommodation			0123456789	150.00
<input type="checkbox"/> Other external services				
<input type="checkbox"/> Movables				
<input type="checkbox"/> Other				

Indirect Expenses

BO: SPA - BU KAIM SF Field Position Norte-01 50.00 %
SPA - BU KAIM Position 001 50.00 %

Perceiver: Region

Region: Islas Baleares Percentage: 100.00

	Alias	Mail	PO	Estimated Amount
<input type="checkbox"/> Movables				
<input type="checkbox"/> External services				
<input type="checkbox"/> Infrastructure				
<input checked="" type="checkbox"/> Other			0123456789	25.00

Seleccionaremos los detalles generales del evento

Id: Event: Year: Description:

Dept. Code: % RC:

G/L Account SAP: AC:

Typology: Demand
Eventos
Enviar profesionales a Congreso sin Estructura Amgen
De cualquier ámbito

☒ Active

☐ Aranesp Nefro
☒ Aranesp Onco
☐ Mimpara
☒ Neulasta
☐ Neupogen
☐ Vectibix

A continuación seleccionaremos los conceptos del evento, indicando proveedor, PO y estimado para cada uno de ellos. Los datos que introduzcamos en costes directos serán visibles en el evento. Desde el evento no se podrá seleccionar ningún concepto que no esté marcado en la definición.

Direct Expenses

Guide:

BO:

☐ To be defined in Sponsorships
SPA - BU KAM Position 001

%

Perceiver Type:


	Alias	Mail	PO	Estimated Amount
<input checked="" type="checkbox"/> Inscription			<input type="text" value="0123456789"/>	<input type="text" value="56.00"/>
<input checked="" type="checkbox"/> Travel			<input type="text" value="0123456789"/>	<input type="text" value="43.00"/>
<input checked="" type="checkbox"/> Miscellaneous			<input type="text" value="0123456789"/>	<input type="text" value="152,465.00"/>
<input checked="" type="checkbox"/> Accomodation			<input type="text" value="0123456789"/>	<input type="text" value="150.00"/>
<input type="checkbox"/> Other external services				
<input type="checkbox"/> Movables				
<input type="checkbox"/> Other				

Pulsando nos aparecerá el histórico de PO y estimados.

Historic POs		
Data	PO	Est. Amount
11/06/2008	0123456789	87.00
30/06/2008	0123456789	152,465.00

Por último nos quedará definir los conceptos indirectos. Estos conceptos serán comunes para todas las colaboraciones inscritas al evento.

Indirect Expenses


BO: 

SPA - BU KAM SF Field Position Norte-01

50.00 %

SPA - BU KAM Position 001


50.00 %

Perceiver:  Region

Region

Isias Baleares

Percentage


100.00 

☐ Movables

☐ External services

☐ Infrastructure

☒ Other

Alias	Mail	PO	Estimated Amount
<div></div>		0123456789	25.00 